ESD-TR-75-353 MTR-3035

A SUMMARY AND FINAL REPORT OF THE RIDS PROGRAM

JANUARY 1976

Prepared for

DEPUTY FOR DEVELOPMENT PLANS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Bedford, Massachusetts

Project No. 6370
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-75-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

Emery F. Boose

EMERY F. BOOSE
Leader MITRE D-92

Keith Handsaker

KEITH HANDSAKER
Project Engineer

FOR THE COMMANDER

John B. Tindall

JOHN B. TINDALL, Lt Colonel, USAF
Acting Director, Technological Planning
Deputy for Development Plans

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ESD-TR-75-353
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): A SUMMARY AND FINAL REPORT OF THE RIDS PROGRAM
5. TYPE OF REPORT & PERIOD COVERED: SUMMARY AND FINAL RPT.
6. PERFORMING ORG. REPORT NUMBER: MTR-3035
7. AUTHOR(s): Digital Avionics Group
8. CONTRACT OR GRANT NUMBER(s): F19628-75-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The MITRE Corporation, Box 208, Bedford, MA 01730
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project No. 6370
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Development Plans, Electronic Systems Division, AFSC, Hanscom Air Force Base, Bedford, MA 01731
12. REPORT DATE: JANUARY 1976
13. NUMBER OF PAGES: 88
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

AIRBORNE INFORMATION DISTRIBUTION SYSTEM
MULTIPLEX BUS
TIME DIVISION MULTIPLEXING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is the final report for MITRE Project 6370, the Radio Information Distribution System (RIDS) program. It summarizes the analytical and experimental work done on this program, describing in detail the experimental multiplex bus which was built, and the demonstrations using AN-numbered avionics which were performed on that bus. A comprehensive set of references for the program is included.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ACKNOWLEDGMENT

This report has been prepared by The MITRE Corporation under Project 6370. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS
AND TABLES

SECTION I

INTRODUCTION

Background

Traditionally, when the U. S. Air Force requires a new weapon system, it relies upon a System Program Office (SPO) which has the responsibility for the procurement of the new system. The SPO requests bids from industry, selects contractors, and awards contracts for the design and development of the weapon system. Often, the new system is uniquely designed because a different requirement must be met or a form factor must be different. The result is that the USAF inventory contains a variety of different and incompatible pieces of avionics equipment, many of which are designed to provide essentially the same function. Each of these systems requires its own inventory of replacement parts and test equipment which may be of no use to other systems, and this results in increasingly high O&M costs.

The U. S. Air Force, recognizing that airborne information distribution systems are beginning this same cycle, and determined to cut O&M costs while increasing flexibility and reliability, is developing MIL-STD-1553 (USAF) for a digital multiplex data bus system (1). The standard can be used on all aircraft to replace hardwired point-to-point cable harnesses for the control and distribution of information throughout the aircraft. All avionics subsystems will couple to the bus via standard interface units in a standard digital format. Figure 1, taken directly from the standard, shows the multiplex data bus architecture as defined in MIL-STD-1553.

During FY ´73, Project 6370, the Radio Information Distribution System (RIDS) program, was initiated at MITRE in support of the Electronic Systems Division (ESD) of the Air Force. The program was an outgrowth of the CNI (communication, navigation, and identification) program which started in the late 60´s and developed into the SEEK BUS (now JTIDS) and the RIDS programs. The general purpose of the RIDS program was to evolve and define preliminary designs for integrating radio information systems (i.e., all the radio, communication, navigation, and identification equipment). Specific tasks included support of the Digital Avionics Study Group at WPAFB, and later, the Digital Avionics Information System (DAIS) program. Some work was also funded directly by the DAIS program in FY ´74 under Project 6540.

The airborne multiplex (MUX) bus described in MIL-STD-1553 (USAF) is a half-duplex system using a twisted shielded pair of

IA-45,908

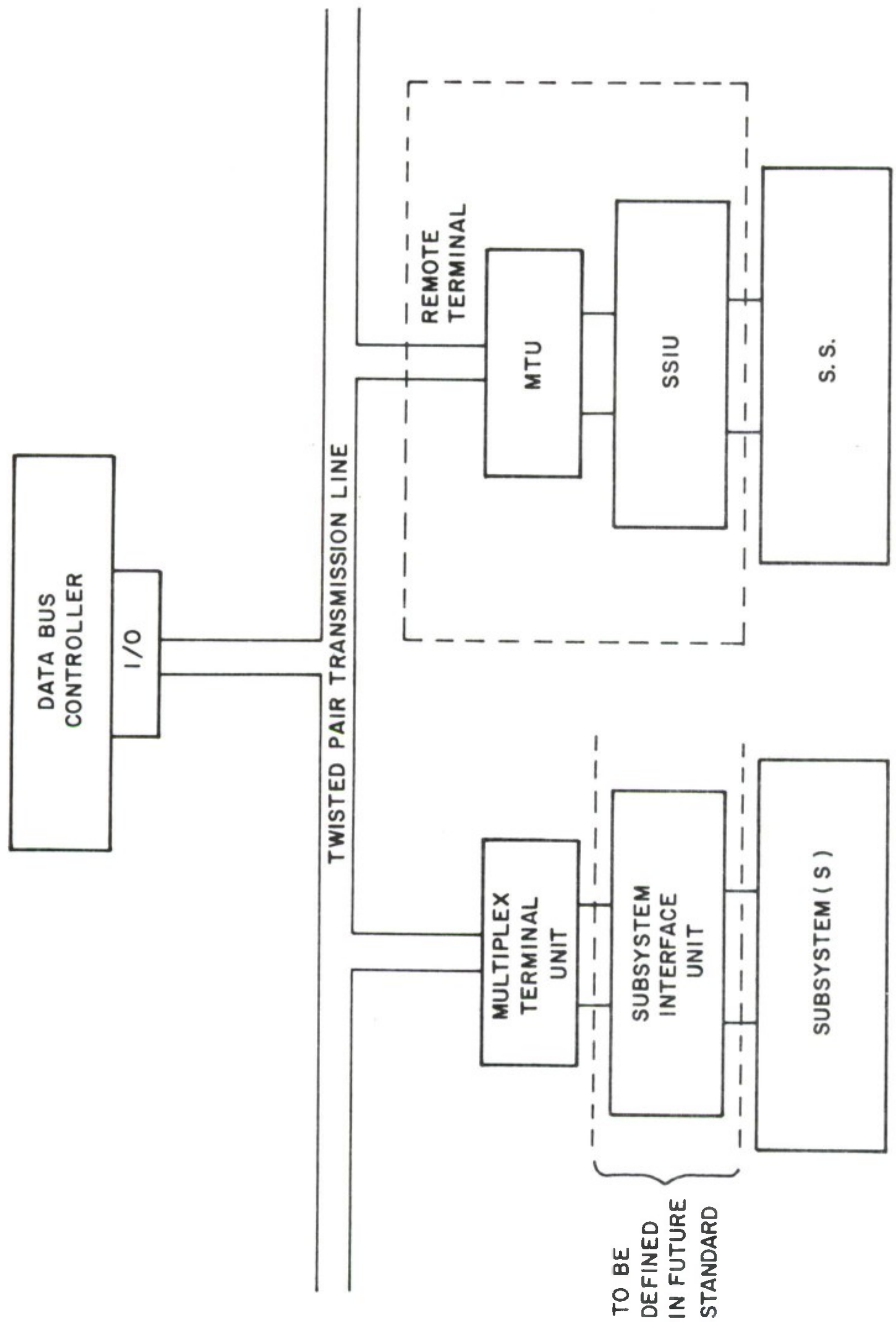


Figure 1 DATA BUS ARCHITECTURE

wires as the transmission medium over which digital data is distributed throughout the aircraft in a command-response mode of operation under control of the central processor. The data rate on the bus is one megabit per second, and the data is coded in Manchester II.

The displays and controls required for the selected radio subsystems will be common, and they will be time shared with other avionics subsystems.

It is intended that all monitor and maintenance functions will be accommodated over the MUX bus through the use of Built-In-Test Equipment (BITE) and a Central Integrated Test System (CITS). This system will also greatly facilitate ground maintenance through the use of common ground check-out and test equipment.

A standard interface unit which incorporates a microprocessor as the principal element in the interface between the avionics subsystems and the bus was investigated under this program. This small programmable computer allows the flexibility required for interfacing with a large variety of subsystems. The microprocessor being used in the investigations is one developed by National Semiconductor.

The RIDS MUX bus system has provided useful inputs for the Air Force effort to develop a military standard for multiplex data buses and has, at the same time, investigated the interface problems which will be met by radio subsystems using a multiplex bus for the first time.

Purpose

The major objective of Project 6370 was to gain experience in the design and operation of MUX bus systems built according to the Air Force standard, to test the feasibility of its concepts for the purpose of providing useful feed-back to the group at WPAFB responsible for the specification of these standards, and to investigate the interfacing of radio systems to the bus. In order to achieve these objectives, a rudimentary experimental multiplex data bus based on MIL-STD-1553 has been designed and built for the purpose of validating standards and specifications, and for providing a vehicle for the demonstration of multiplex data bus concepts.

Scope

The RIDS MUX Bus does not require the full capabilities called for in the Standard in order to demonstrate multiplex data bus concepts. The word formats, message types, data rates, and functions performed comply with the Standard. However, in the interests of economy, the number of words per message, the number of available remote terminal (RT) addresses, and the number of permissible subaddresses have been reduced from the 32 per terminal called for by the standard to 16. Moreover, the number of remote terminals actually built for the RIDS MUX bus is only two. One of these remote terminals interfaces with the subsystems through a hardwired multiplexer, and the other RT uses a programmable microprocessor as its subsystem interface.

The subsystems, which are controlled by means of digital signals time division multiplexed over the bus, consist of various representative types of equipment which were available. These include an AN/ARC-50 UHF set, a gyro-compass indicator, an air speed indicator, a radio altimeter indicator, a fuel gauge, a pressure gauge, and various lights. Some of these are driven by synchros, some are controlled by switches, some by dc voltage levels, some by coded dc voltages, and some by analog voltages. Signal conversion circuits were designed and built to convert these signals to digital data for transmission over the bus, and to reconstitute digital data back into the necessary driving signals to operate the equipment. Although the subsystems used in conjunction with the RIDS MUX bus for demonstration purposes may not be equipment which would ever be used in an aircraft configuration, their signal requirements are representative.

Since the RIDS MUX bus was intended only as a vehicle for investigating multiplex bus concepts specified in the standard, no special attempt was made to maximize space, weight, and power savings. The circuits are all designed and built as laboratory bread-boards.

The controller is a DEC PDP-9 computer. This is hardly the sort of equipment one would install in a lightweight fighter plane, but it is a computer which was available and suitable for the RIDS demonstrations.

The RIDS MUX bus system does demonstrate controller-to-RT, RT-to-controller, and RT-to-RT standard message transfers. It demonstrates the control of a variety of avionics equipment over a single twisted shielded pair of wires. It demonstrates the inherent flexibility of the multiplex data bus concept for subsystem retrofitting and ease of system reconfiguration for various types of missions.

The RIDS effort has not only produced a versatile tool for testing and evaluating MUX bus concepts and for demonstrating the operation of various system configurations, but it has also provided expertise in all the various aspects, both hardware and software, relating to MIL-STD-1553 type of multiplexed digital data buses.

The components which make up the experimental MUX bus are described in Section II. These include the transmission medium, the remote terminals, controller, and bus controller interface unit. Section III describes the avionics equipments which are connected to the bus along with their unique interfaces. Section IV covers the details of system operation including the system software which makes the operation possible. Section V discusses briefly some of the problems which were encountered in making the bus operational. Section VI reviews the results and Section VII states the recommendations and conclusions which result from the RIDS program.

SECTION II

SYSTEM COMPONENTS

General

This section describes the various components which make up the multiplex data bus system as configured in the laboratory. It does not include descriptions of the various avionics subsystems which interface with the MUX bus system in the laboratory demonstration hardware, nor does it describe the signal conditioning circuits which were designed to convert signals to and from the avionics equipment to MUX bus compatible signals. These will be described in Section III.

The system components described in Section II include the following:

- Bus (the transmission medium)
- Controller
- Bus Control Interface Unit (BCIU)
- Multiplex Terminal Unit (MTU)
- Subsystem Interface Unit (SSIU)
- Multiplexers.

Bus

The transmission medium used for the RIDS MUX bus is a twisted-shielded pair of wires having characteristics equivalent to standard RG 108A cable. The actual wire used is Haveg Industries type LE-572-0003/0002 cable, a lightweight version of the RG-108A, which is the one recommended for use in multiplex bus applications on the B-1 aircraft by North American Rockwell. This cable is in accordance with MIL-STD-1553 (USAF). Laboratory measurements (2) on samples of this cable showed it to have the following average characteristics:

$C = 21.8$ pf per foot
$L = 0.109$ μh per foot
$R = 0.0288$ ohms per foot
$Z_0 = 71$ ohms (characteristic impedance)

Function

The function of the bus is to transmit digital data throughout the aircraft between the bus controller and various avionics subsystems. The data bus functions asynchronously in a command/response mode, and transmission over the bus is half-duplex.

Control of traffic on the bus resides solely with the controller which initiates all information transfers.

Configuration

Figure 2 shows the data bus interface. The bus itself is terminated at both ends (not shown in the figure) with a resistor approximately equal in value to the characteristic impedance of the line. Stubs, of the same cable as the main bus, are connected at various stations along the bus, and are coupled to the bus by means of a transformer. As shown in Figure 2, isolation resistors are inserted in each leg of the stub near the stub-bus junction to prevent shorts which might occur on the stubs from shorting out the main bus. The stubs are used as drops from the bus to various "station" locations in the laboratory. Remote terminal equipment can be simply uncoupled from its stub at one station, moved to another station, coupled to the stub at that location, and be ready to operate. This demonstrates the ease with which the system can be reconfigured.

Data

The information flow on the bus comprises messages made up of command words, data words, and status words. Each word is composed of a sync pulse of three microseconds duration followed by sixteen one-microsecond data bits plus a one-microsecond parity bit. The total word length is twenty microseconds long. A message on the RIDS bus can vary from a minimum of three words (a command word, a data word, and a status word) to a maximum of twenty words (two command words, sixteen data words, and two status words). As was noted in Section I above, this differs from the thirty-two data word message capability called for in MIL-STD-1553 (USAF).

Three modes of information transfer are permitted on the RIDS bus. These are: controller (CTLR) to remote terminal (RT), RT to CTLR, and RT to RT. Each is described in detail in Section IV in the subsection dealing with message formats.

Data is transferred over the bus in serial digital pulse code modulation form, and the data code on the bus is Manchester II bi-phase level in which a logic "one" is transmitted as a bipolar coded signal 1/0 (i.e., a positive pulse followed by a negative pulse, each of one-half microsecond duration), and a logic "zero" is a bipolar coded signal 0/1 (i.e., a negative pulse followed by a positive pulse). A transition through zero occurs at the midpoint of each bit time. Figure 3 shows a graphic illustration of Manchester coded data. The sync pulses are transmitted at one third the bit rate of data pulses and are of two types. The command and status sync pulses are identical, and they each consist of a bipolar

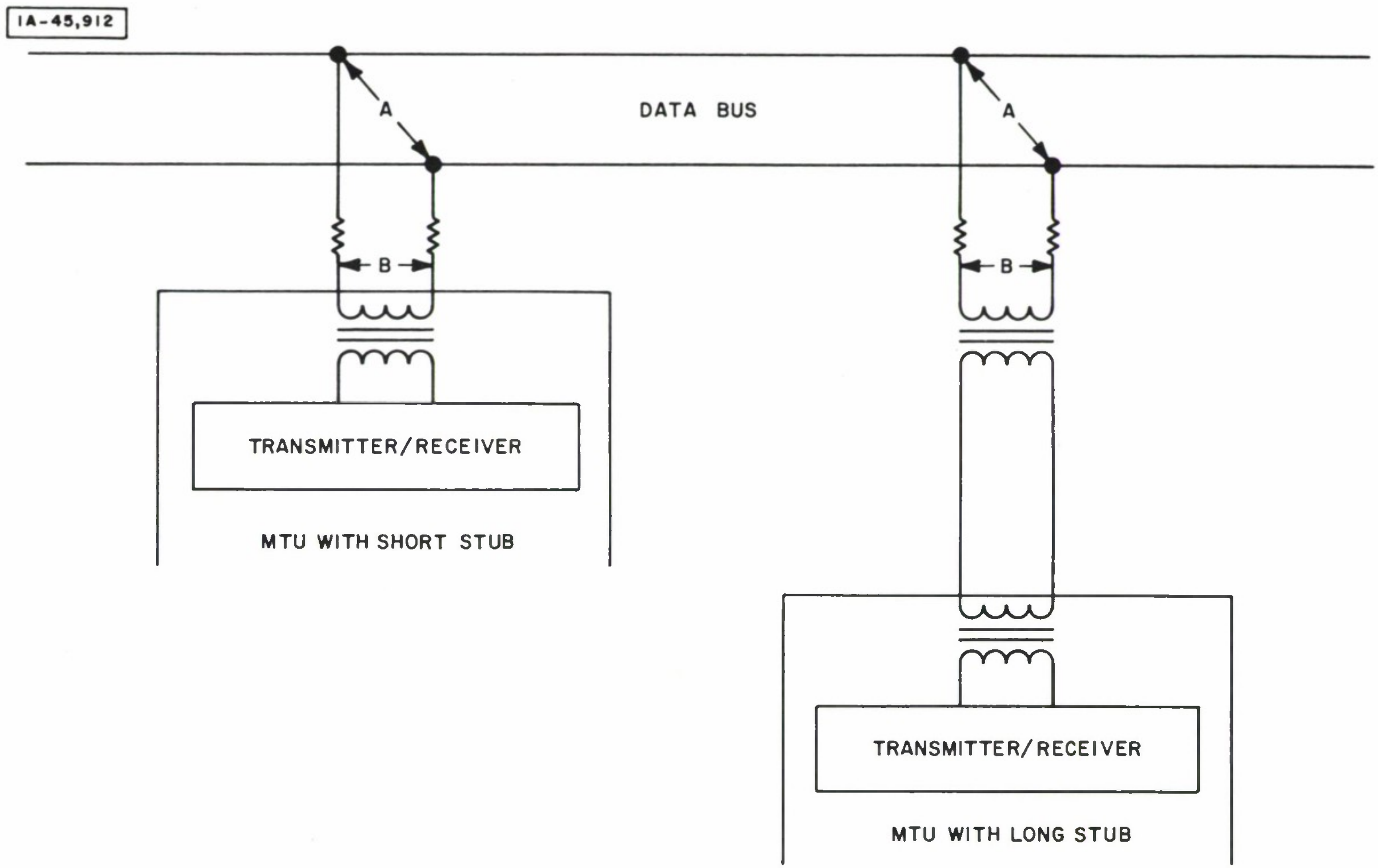
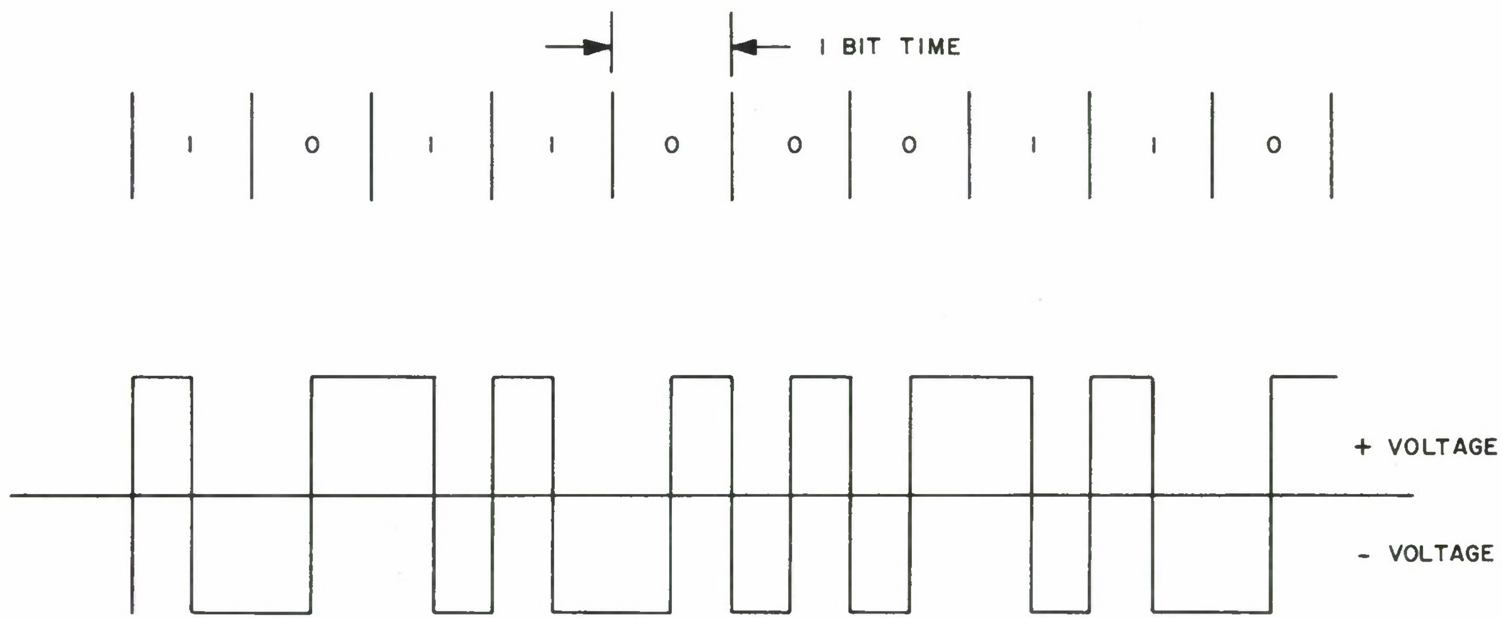


Figure 2 DATA BUS INTERFACE

IA-45,911



NOTES: MANCHESTER II BI-PHASE LEVEL

"1" REPRESENTED BY PLUS/MINUS

"0" REPRESENTED BY MINUS / PLUS

Figure 3 DATA CODE

coded signal 1/0 in which a positive 1-1/2 microsecond pulse is followed by a negative 1-1/2 microsecond pulse. The data sync pulse is the same duration as the other sync pulses, but is a bipolar coded signal 0/1.

The bus operates as a balanced line, and as such provides a high degree of common-mode rejection of noise picked up from external sources. The shielding on the bus is grounded to further protect against noise pick-up. For more detailed information on the bus, see Reference 2.

Controller

The bus controller is a key part of the data bus system. It initiates all messages, including those sent from one remote terminal to another, by means of command words which it transmits. These command words are each addressed to a particular remote terminal which then takes action based on the instructions contained in the command.

Function

The function of the RIDS controller is to regulate the flow of traffic on the bus. The controller initiates each message by means of a command word addressed to a particular remote terminal, which in turn responds to the information contained in the command word. The word formats are explained in detail in Section IV of this report. The controller can send up to sixteen data words plus the command word to either of the two remote terminals (each RT can be assigned any one of sixteen different addresses), it can request up to sixteen data words to be transmitted from the RT, and it can command one RT to receive data (up to sixteen words) and the other to transmit. The RTs return a status word after receiving the number of data words specified in the command word or before transmitting data words if it has been commanded to transmit. The controller checks the receipt of the status word from the RTs.

Equipment

The equipment used for the RIDS MUX bus controller is a Digital Equipment Corporation (DEC) PDP-9 computer which was available. The computer interfaces with the data bus through a Bus Controller Interface Unit (BCIU).

The PDP-9 is a single address, fixed word length (18 bits), parallel binary general purpose computer. The system has 8192 words of core memory storage, paper tape input and output, console teleprinter keyboard input and printer output at 10 cps (ASR-33), and a magnetic tape transport. A Sanders 720 CRT Display system is also coupled to the computer through some special logic circuitry so that the computer can print out alpha-numeric information on the CRT display as well as on the teleprinter. The Sanders unit is also equipped with a keyboard through which data can be inserted into the system, displayed on the CRT, and fed into the computer. Figure 4 is a photograph showing the PDP-9 computer which is being used as the RIDS MUX bus controller. The picture shows a portion of the Sanders display unit and the teleprinter on the extreme left-hand side.

Bus-Controller Interface Unit

The Bus-Controller Interface Unit (BCIU) functions as the interface between the controller and the bus. In the photograph of Figure 4, the BCIU is the rack of equipment shown to the right of the magnetic tape unit. It is obvious from the picture that no attempt was made at saving space and weight.

The BCIU consists of three distinct subunits. These include signal conversion circuits for converting PDP-9 logic signal levels to TTL compatible signals, controller interface circuits for performing all necessary handshaking functions with the controller, and bus interface circuits. The controller interface circuit and the bus interface circuit interface with each other and pass parallel data and control signals both ways. The bus interface unit is a modified Multiplex Terminal Unit (see MTU subsection).

The RIDS BCIU performs several important functions. First of all, it is necessary to convert from the DEC PDP-9 negative logic signals to the TTL compatible signals used in the rest of the RIDS system. Next, the BCIU performs all of the necessary handshaking with the PDP-9 computer. And finally, it interfaces with the MUX bus, converting digital data received in parallel from the computer to serial data, generating the proper sync pulse header, and encoding the data into Manchester II code for serial transmission onto the bus. On receipt of data from the bus, the BCIU decodes the Manchester II data, converts from serial to parallel, stores the message in memory and notifies the computer that a message has been received. When the computer is ready to receive, the message is transferred one word at a time in parallel from the BCIU to the computer. During this transfer, signal conversion must again take place to translate from TTL logic levels to computer compatible signal levels.

Multiplex Terminal Unit (MTU)

The MTU serves as the interface between the multiplex data bus and the subsystem interface unit (SSIU). The SSIU is described in

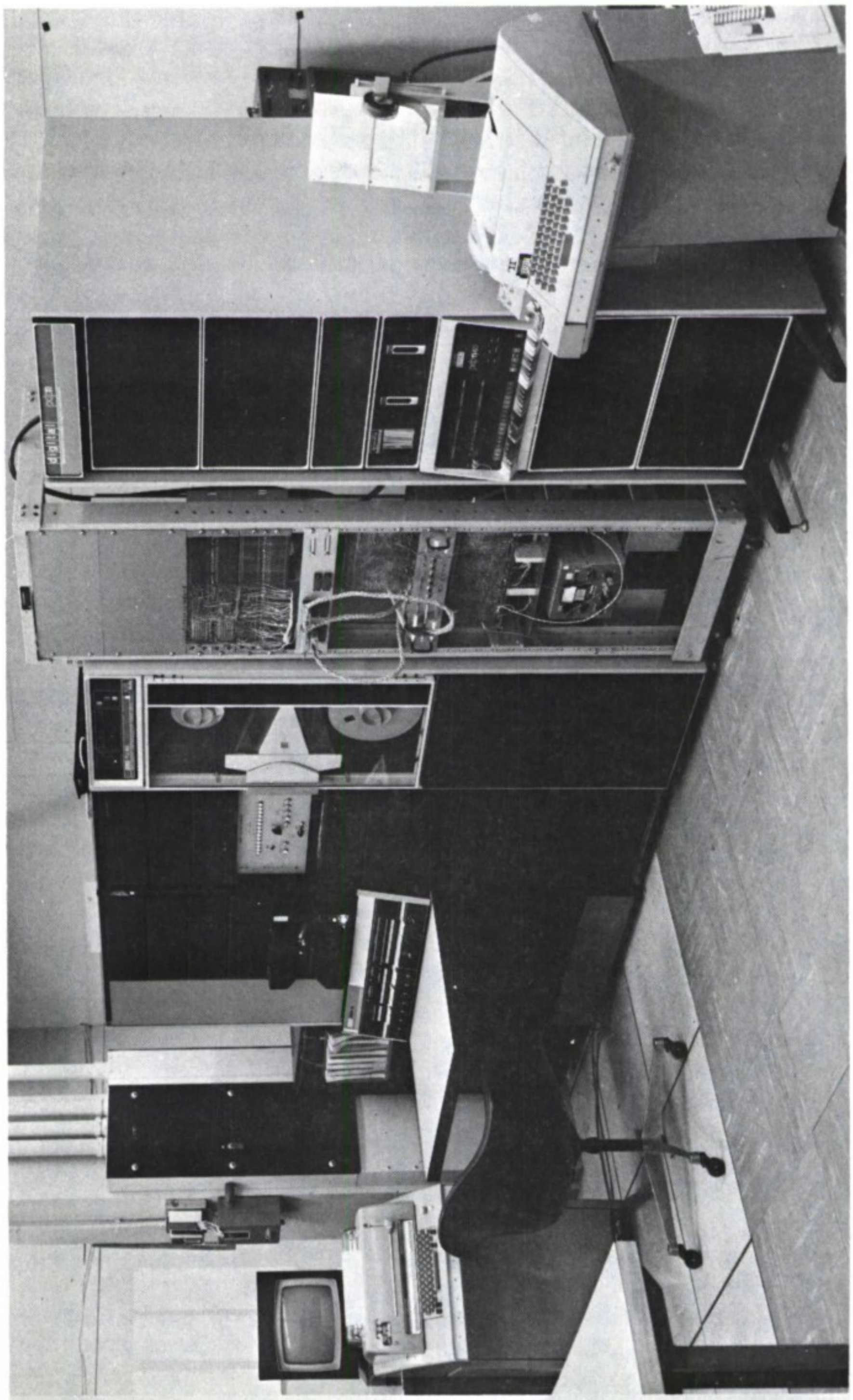


Figure 4 RIDS MUX BUS CONTROLLER AND BCIU

the next subsection. The MTU and SSIU, together, constitute a remote terminal (RT).

Functions

The RIDS MTU consists of three main sections. These are: (1) the transmit-receive-and timing (TRT) section, (2) the control (CTL) section, and (3) the switching (SW) section. The functions of each of these sections is described below.

TRT Function. The functions of the TRT section of the MTU are to receive Manchester II coded data from the bus at a 1 megabit per second rate, translate this data into NRZ binary coded data at signal levels compatible with TTL circuits, check incoming messages for a valid sync pulse at the beginning of each incoming word, distinguish between command and data sync pulses, and notify the CTL section which type sync pulse has been received. The TRT section generates all timing signals required in the MTU. On outgoing messages, the TRT section generates a status or a data sync pulse at the beginning of each transmitted word, it encodes the messages to be transmitted into Manchester II data, and it transmits this data onto the bus at the proper signal voltage, power level, and rate.

CTL Functions. The functions of the CTL section include performing all necessary logic for the control and sequencing of all events which take place in the MTU. The MTU continuously monitors the bus and checks all incoming sync pulses. When a command sync has been recognized, the CTL section checks the message address (first five information bits following the sync pulse) and responds only to those messages which carry the address assigned to it. All other messages are ignored.

The CTL also recognizes the difference between a transmit command and a receive command, and responds accordingly. It also responds to the data word count information contained in the received command word, and either sends or receives the correct number of data words. It checks for bit parity on incoming data, and generates the proper parity bit for outgoing data.

When commanded to receive, the CTL section controls the receipt and storage of the incoming message from the bus, and if no parity error has been detected in any of the words, it controls the transfer of the message (including the command word) to the SSIU.

When commanded to transmit, the CTL section passes this information to the SSIU and then controls the receipt and storage of the designated number of words from the SSIU, the transfer, one at a time, of each word from storage in the memory to the I/O register, and the serial clocking out of each word from the I/O register to the transmit circuits in the TRT. All of the switching functions of the SW section are controlled by the CTL section.

The CTL section also assembles and controls the transmission of the status word at the end of every message, and it controls the transmission of the data words in the message. Finally, the CTL section monitors the bus at all times for "bus busy" conditions, and prohibits the MTU from transmitting when the bus is busy.

SW Function. The SW section of the MTU performs all serial-to-parallel and parallel-to-serial conversions of message words, storage of the complete message prior to transfer to either the TRT section for transmission onto the bus or to the SSIU, and all data switching functions of the MTU. The control of all these functions resides in the CTL section.

Equipment

The RIDS laboratory demonstration bus has two remote terminals. These are shown in the picture of Figure 5. One of these RT units is on the table, and the other is mounted on the rack. A close-up of the unit on the table is shown in Figure 6. The MTU and SSIU circuit boards are shown in their upright position for easy access during circuit testing; they can both be folded down to lie flat (one below the other) inside the cabinet. The MTU circuit board is behind the SSIU board.

The MTU consists physically of three circuit boards mounted side by side in a single frame 16-1/4 inches by 7-1/4 inches. Each board contains one of three MTU functional sections. The boards have wire-wrap pins on one side and transistor-to-transistor logic (TTL) circuit chips mounted on the other. Circuit connections are made on the wire-wrap side by means of insulated wire leads. Switches on the front panel of the RT cabinet permit the assignment of any one of 16 addresses to the MTU. The controller routes messages to a particular RT unit by utilizing the RT's assigned address in the command word which initiates the message transfer.

Subsystem Interface Unit

The Subsystem Interface Unit (SSIU) is one of the two basic components of the remote terminal (the other is the MTU described in the preceding subsection of this report). The SSIU serves as the interface between the MTU and the avionics subsystems. Figure 7 shows a block diagram of the remote terminal architecture specified by MIL-STD-1553 (USAF). (This will probably be modified by a revision of the standard now under consideration.)

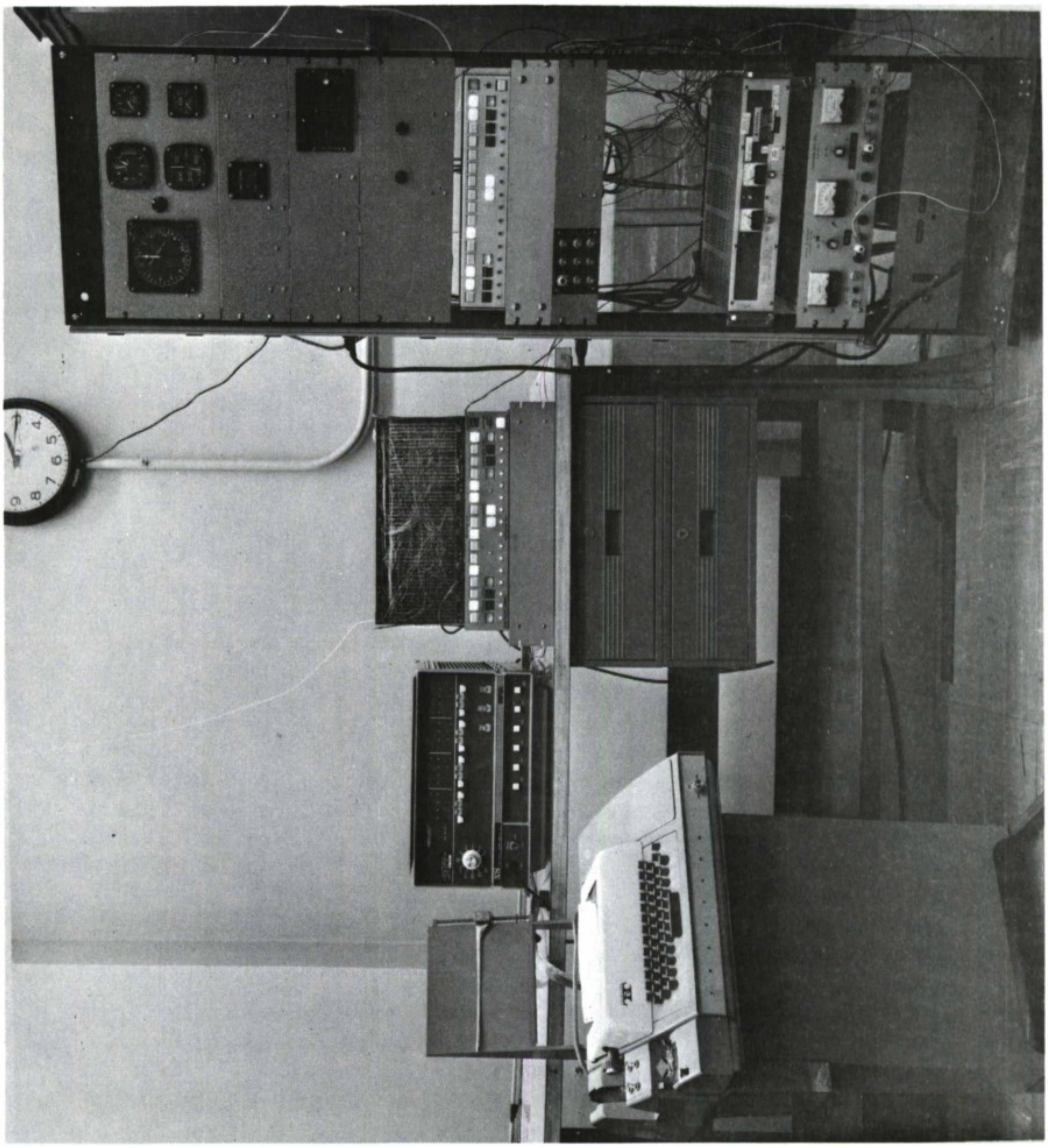


Figure 5 REMOTE TERMINALS

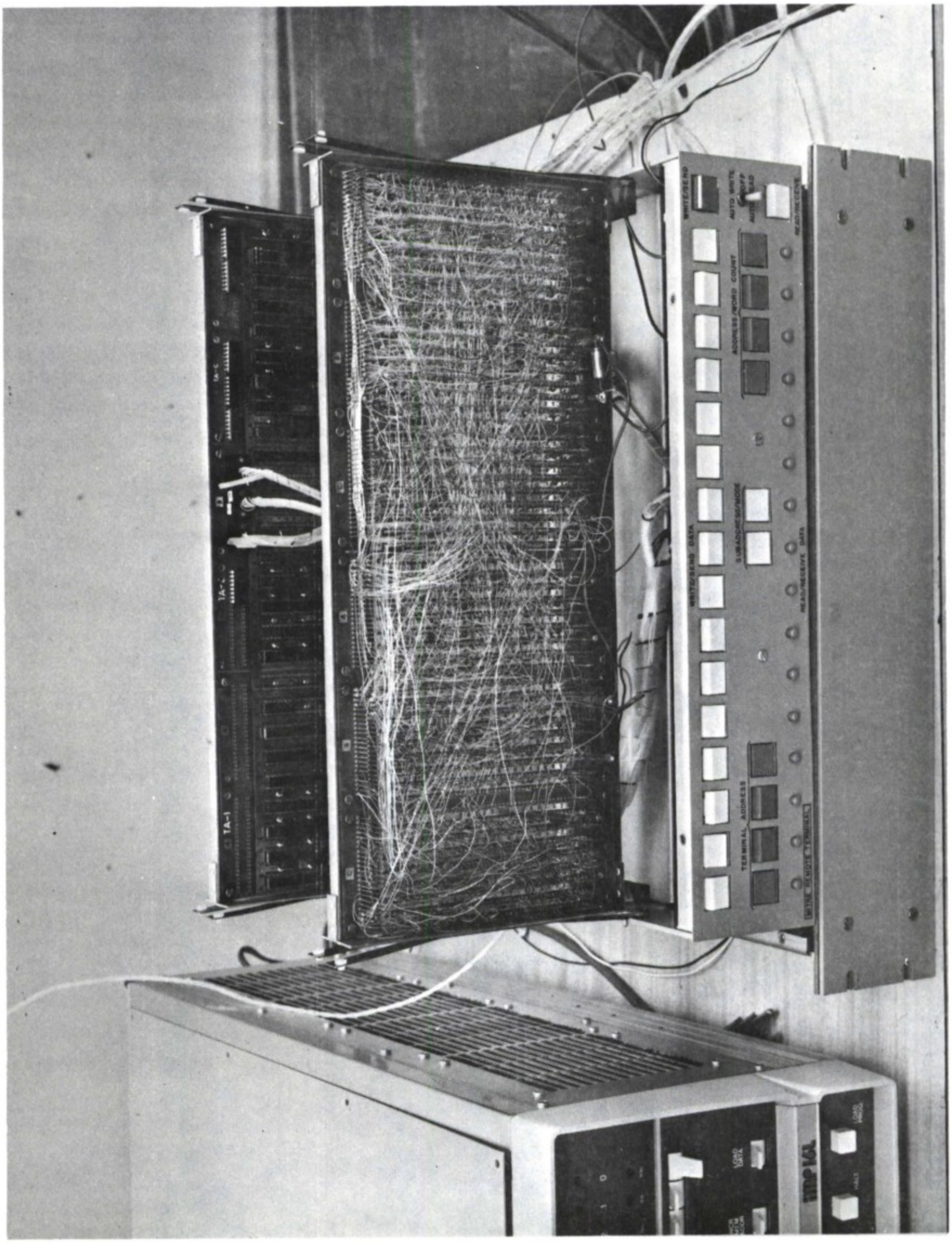


Figure 6 CLOSE UP OF REMOTE TERMINAL

IA-45,910

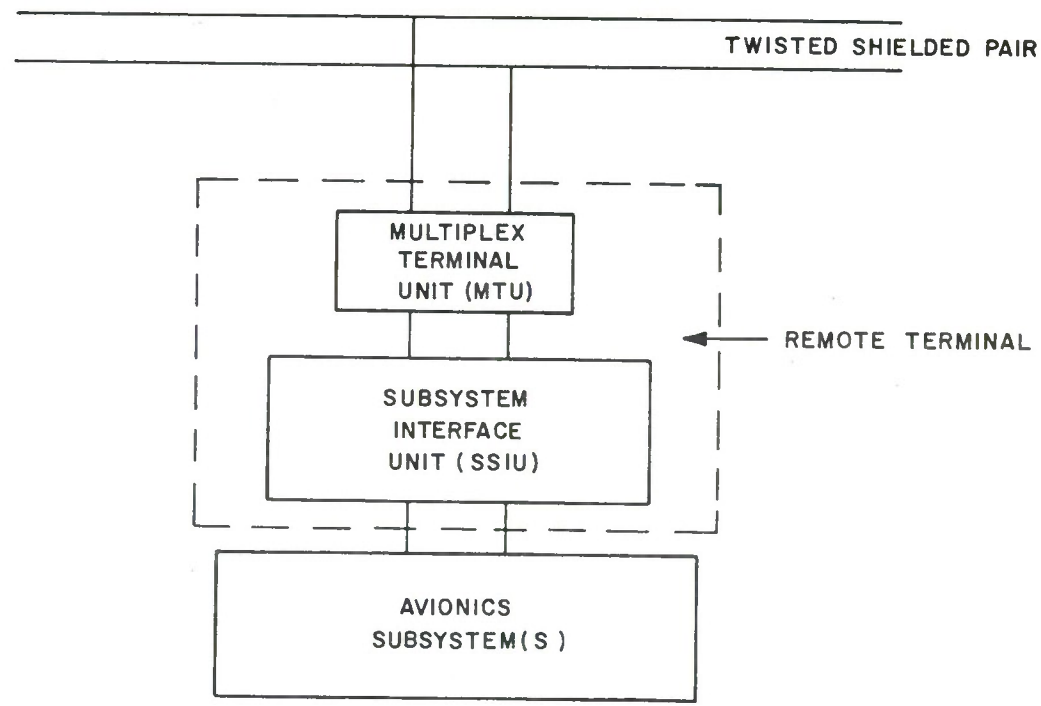


Figure 7 INTIALLY PROPOSED REMOTE TERMINAL ARCHITECTURE

The SSIU must perform several tasks in the multiplex bus system. First, it must interact with the MTU, providing data to be sent to the controller on request or storing data from the controller via the MTU. The second task assigned to the SSIU is the collection of data from and distribution of data to the avionics subsystems at predefined times and in a predefined order. The rates at which data is transferred between the remote terminal and the avionics subsystems must satisfy the sample data rate requirements of the subsystems being serviced. A third task assigned to the SSIU is the conditioning of input and output signals between the remote terminal and the avionics subsystems. These signal conditioning functions include analog-to-digital, synchro-to-digital, parallel-to-serial conversions and their inverse operations.

Because of the unique characteristics of each of these tasks, the SSIU as developed for RIDS was subdivided into smaller subsystem components, with each component handling a specific task. The "SSIU" as described in this section of the report includes only that portion of the total SSIU which performs the first of the tasks described above, i.e., the interaction with the MTU interface. The second task, that of collecting data from and routing data to the avionics subsystems, is described later in this paper in the subsection entitled "SSIU Multiplexers". The signal conversion task is subsystem dependent, and therefore is treated separately from the SSIU and is described in Section III of this report.

Functions

One of the basic functions of the SSIU is to satisfy the interface requirements of the MTU. In this capacity the SSIU must store data sent to it by the controller via the MTU for use by the avionics subsystems; and conversely, it must store data sent to it by the avionics subsystems, and provide this information to the controller (again via the MTU) when requested to do so. In order to perform this task, the SSIU interfaces with the MTU, providing not only half-duplex data flow between the two units, but also performing several handshaking functions as well. The RIDS SSIU-MTU interface circuits were designed to function in accordance with the requirements specified in MIL-STD-1553 (USAF). As previously mentioned, the SSIU must also provide data to and accept data from the avionics subsystems at data rates dictated by the nature of the subsystems. Since the SSIU memory access rates dictated by the avionics subsystems are asynchronous with respect to the access rates required by the controller (via the MTU), some form of buffered storage is necessary for decoupling these two processes. An additional requirement levied on this buffered storage is that it must avoid a lock-out situation from occurring when both processes demand access to the same storage cell simultaneously.

To satisfy these requirements, a two-port memory was developed for the SSIU. This memory circuit plus the associated control and interface logic constitute that piece of equipment which is known as the RIDS "SSIU". Throughout the remainder of this document, the term "SSIU" will refer only to this portion of the remote terminal unless otherwise stated. Figure 8 shows a block diagram of the RIDS remote terminal (the blocks shown within the dotted line). The RT includes the MTU, SSIU, RT Test Set, and the SSIU Multiplexer. In RIDS, the signal conditioning circuits are grouped with the avionics subsystems for the sake of convenience since they are so closely interdependent.

Figure 9 shows the basic components of the SSIU. Both the "A Port" (or MTU data line bus) and "B Port" (or SSIU Multiplexer data line bus) use three-state output devices which permit half-duplex data transmission over the buses. The SSIU Test Set has the operational capability of loading data into or reading data from any location in the SSIU memory. This provides the capability for monitoring any selected data location. In the test condition, an off-line mode, the SSIU Test Set is capable of entering data into the "A Port" of memory in a format compatible with the MTU/SSIU interface. This capability provides a means of testing the SSIU circuits when the MTU and bus controller are not available.

Equipment

The SSIU memory, control, and timing logic circuits were implemented by means of MSI-TTL packages mounted on two circuit boards and interconnected with wire leads. The two circuit boards are mounted side-by-side in a 16-1/4 x 7-1/4 inch frame having spring loaded pins which hold it in place in the remote terminal enclosure. Figure 6 shows both the SSIU and the MTU circuit board frames in their upright positions to provide easy access for testing. The SSIU frame is the one in front. The MTU boards have their MSI circuit chip sides showing, and the SSIU has its wire-wrap side towards the camera. In this experimental configuration, no attempt was made at minimization. The SSIU boards contain not only the SSIU memory, timing, and control circuits but also the SSIU test set logic and some of the interface logic required for the "B Port" to SSIU multiplexer task.

All data bus and control line interfaces between the SSIU and MTU are via 14 and 16 line cables and cable connectors. The switches and lights servicing the SSIU test set are physically mounted on the front panel of the remote terminal unit (see Figure 6) and are connected to the SSIU "B Port" by means of 14 pin cable connectors. The off-line "A Port" test set function is achieved by means of a multi-layer wafer switch mounted on the rear panel of the

IA-45,909

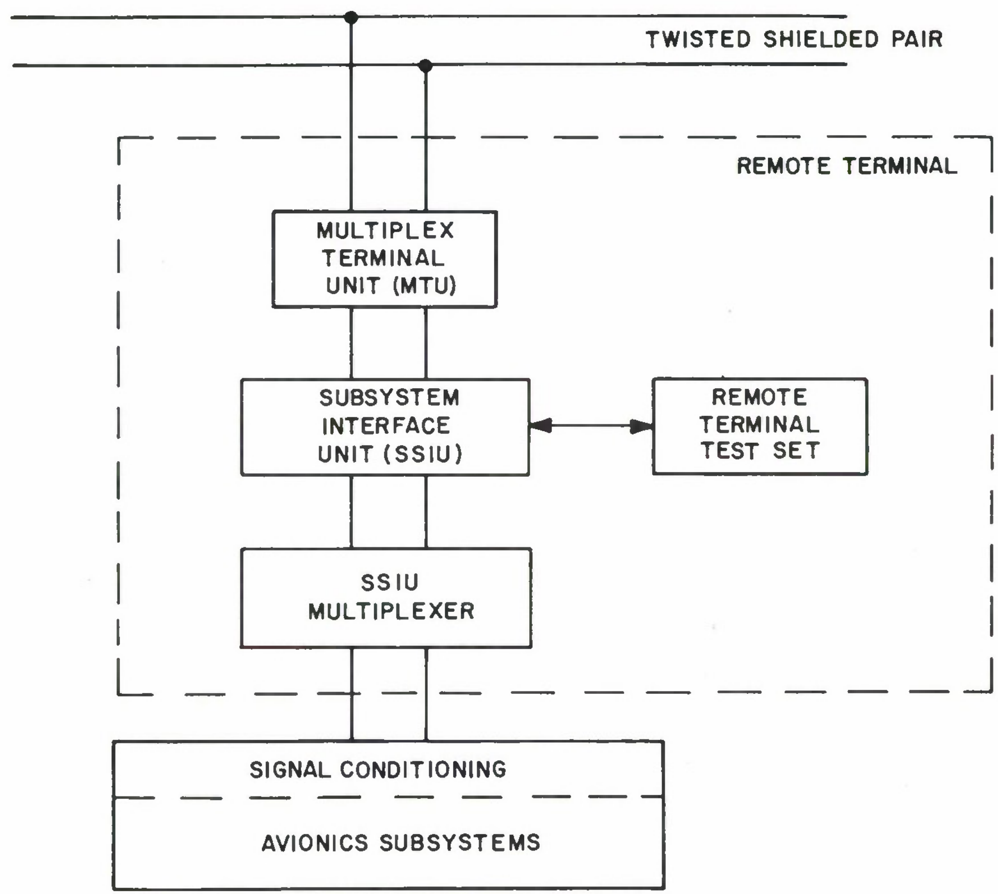


Figure 8 MECHANIZED REMOTE TERMINAL ARCHITECTURE

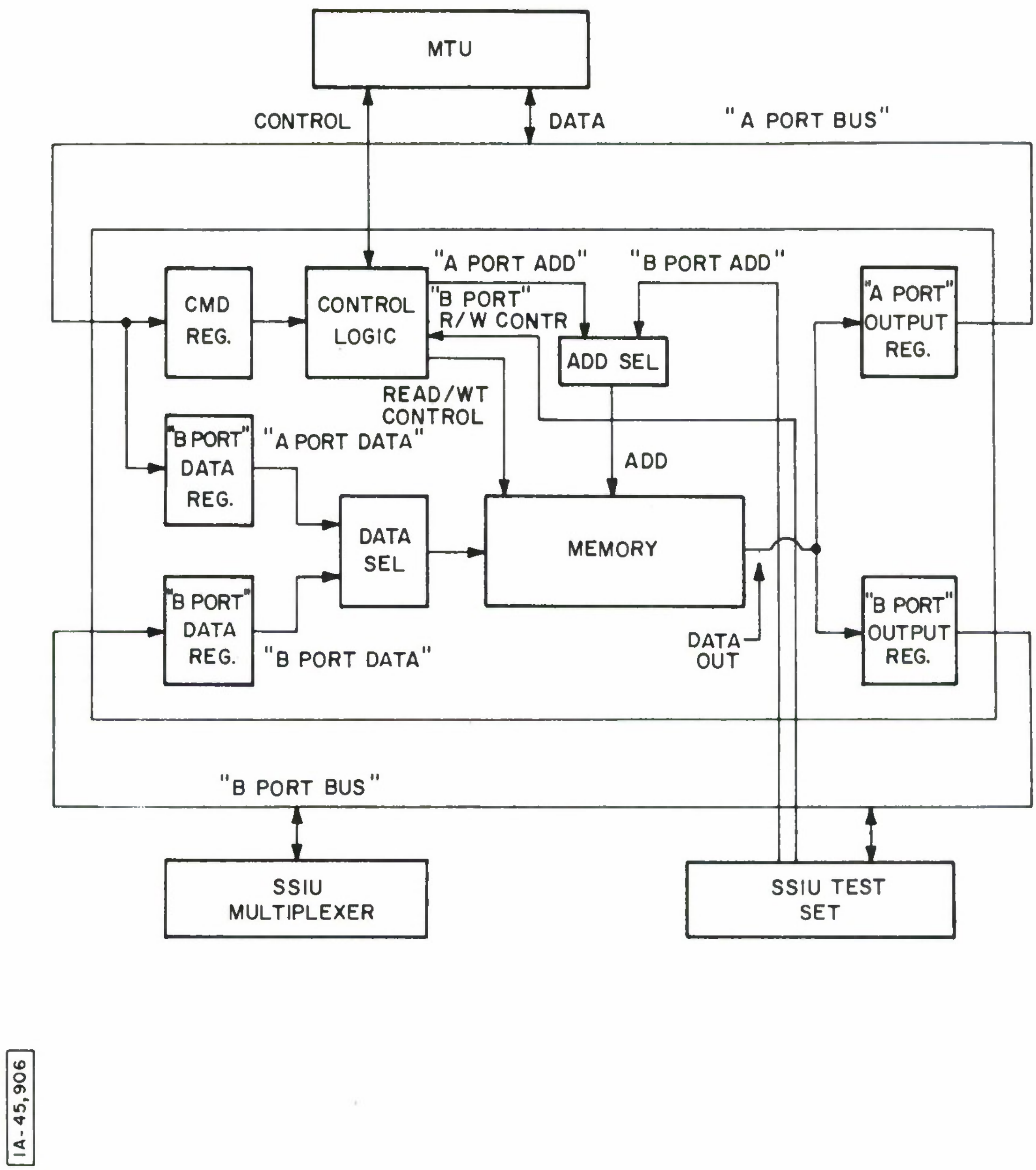


Figure 9 SSIU BLOCK DIAGRAM

RT unit. In the test position, this switch decouples the MTU lines from the "A Port" data bus and substitutes the Test Set data lines.

Because of the experimental nature of this design, several deviations were made from the MIL-STD-1553 specifications (see Section I) for the sake of economy. The standard specifies that the SSIU shall be capable of handling up to 32 subaddresses of up to 32 words each. And each subaddress would be capable of both transmitting and receiving. This full capability would require 2048 16-bit words of memory. Due to the expense of dynamic memory, only 4 subaddress planes of memory were implemented. Because 4-bit decoders were more readily available than 5-bit decoders, the word count capacity was limited to 16 instead of 32 as called for in the standard. Demonstration of MUX bus system operational concepts specified by the standard has not been impaired by the reduction in capacity implemented in RIDS.

Multiplexers

Although technically a part of the SSIU, the multiplexers are treated separately in this document because the multiplexer in one RT is a simple hardwired unit while in the other RT it is a much more sophisticated programmable minicomputer. This great difference between the two units warrants a separate subsection in this document for the multiplexers, and each one is described separately in the following paragraphs.

Hardwired Multiplexer

As was mentioned above, the multiplexer in one of the RIDS remote terminals is a simple hardwired unit. This unit is in essence a sequential switch that connects the SSIU "B Port" data bus (16 parallel lines) to each associated avionics subsystem in a fixed, pre-determined sequence. The multiplexer contains a timing and control circuit, read buffers, and write buffers. These are described below.

Timing and Control Circuit. The Timing and Control Circuit for the Hardwired Multiplexer controls the timing and sequencing of the access intervals between the "B Port" data lines and the transmit and receive buffers which interface with the avionics subsystems. These intervals are synchronized with the SSIU 1 MHz "B" clock, so that the avionics subsystems can be accessed by the remote terminal only during the "B Port" half cycle time slot (see subsection on the SSIU above). Counters divide the "B" clock down to a sequential cycle of 128 enable pulses repeated 7812.5 times a second. Of these 128 sequential pulses, the even numbered ones are all "anded" together to form a string of pulses synchronized with every other 1 MHz "B" clock pulse. These pulses are in turn "anded" with the output of a comparator which compares the output of the counter with the subaddress and word address called for by the manual switches on the front panel of the remote terminal unit. The output of this last "and" function is therefore only "enabled" during the clock pulse associated with the SSIU memory address location requested by the manual switches which have been actuated on the RT front panel. This enable pulse triggers the "Read" clock pulse synchronizer in the SSIU test circuit, and it is thus possible by means of the front panel switches to read out the data stored in any one of the word locations in the SSIU memory. The selected word is displayed on the RT front panel by 16 readout lamps.

The odd-numbered pulses in the sequence of 128 are used for sequentially enabling up to 32 read and 32 write buffers which couple the 16 "B Port" data lines to and from the avionics subsystems. Pulses 1, 5, 9, 13, etc., enable read buffers, and pulses 3, 7, 11, 15, etc., enable the write buffers. Additionally, if all of the write enable pulses are not required for use by the avionics subsystems, the unused ones may be routed to the SSIU test circuit to permit writing into the SSIU memory from the front panel (but only into those memory locations represented by these write enable pulses. This prevents writing into the same memory location by both the avionics equipment and the front panel.)

Write Buffers. The write buffer circuit board is wired to accept up to eight 16-bit 3-state buffers, and the timing and control unit can control up to four of these write boards for a total capacity of thirty-two 16-bit write buffers. The inputs of these buffers are connected to the avionics subsystem outputs, and the three-state outputs of the buffers are connected to the common 16-line SSIU "B Port" data bus. These buffers are all strobed sequentially 7812.5 per second in synchronism with the SSIU "B" clock. The strobe causes the buffer to switch out of its high impedance output state and allows the word stored in the buffer to be transferred into the SSIU memory via the "B Port" data lines. The removal of the strobe pulse returns the buffer output to the high-impedance state and at the same time clocks the next data word from the associated avionics equipment into the buffer. The high impedance state of the buffer outputs prevents the buffers from loading down the common data bus when they are not writing data into the SSIU memory. More than one write buffer can be assigned to the same avionics subsystem sensor if a sample rate greater than 7812.5 per second is required for accuracy. Only as many write buffers as are required need be installed in the "write" circuit cards. However, the sequential sampling time slots are fixed and are reserved for each of the 32 allowable buffers whether they are used or not.

Read Buffers. The read buffers are 18-bit latches, two bits of which are not used. The read buffer circuit boards are wired to accept eight of these 18-bit buffers, and the timing and control circuit is capable of handling up to four of these boards for a total of thirty-two read buffers. These buffers are not three-state devices, but merely latches whose outputs are always available for readout. The inputs of these buffers are indirectly connected to the common SSIU "B Port" data lines through driver circuits common to all eight buffers on the same board. The timing and control circuit strobes each read buffer during its designated time slot, causing the word selected from the SSIU memory and placed on the "B Port" data bus during that particular time interval to be read into the buffer. This word is then available to the associated avionics subsystem until such time as it is updated by another word from the SSIU memory (a minimum of 128 microseconds later). As was the case for the write buffers, more than one read buffer can be assigned to the same avionics sensor if the sensor requires an update rate greater than 7812.5 per second. Only as many read buffers as are required need to be installed, but the sequential sampling time slots are fixed and are reserved for each of the 32 allowable read buffers whether they are used or not.

Microprocessor

The SSIU multiplexer associated with the second RIDS RT unit consists of a programmable National Semiconductor IMP-16L microprocessor. This multiplexer is more complex than the hardwired unit, but it is also more flexible and can perform many complex functions.

The basic task of the microprocessor multiplexer, however, remains the same as the hardware multiplexer. This task consists of periodically sampling and disseminating data from the "B" Port of the SSIU's memory to the input and output registers of the multiplexer unit. Unlike the hardware multiplexer, the data handling sequence is not fixed but is executed under software program control and is readily alterable. The insertion of the microprocessor into the data transfer path between the I/O registers and the SSIU's memory also allows the designer to take advantage of the computational power of the microprocessor. This capability can be used in a variety of ways to reformat the data to be transferred or dynamically alter the sampling sequence based on the data content. This section of the report will not dwell on the use of this capability but will briefly describe the implementation of the microprocessor into the multiplexer design.

Timing and Control Circuits. Figure 10 shows a block diagram of the microprocessor multiplexer incorporated into the remote terminal design. The multiplexer system bus is tied to the SSIU

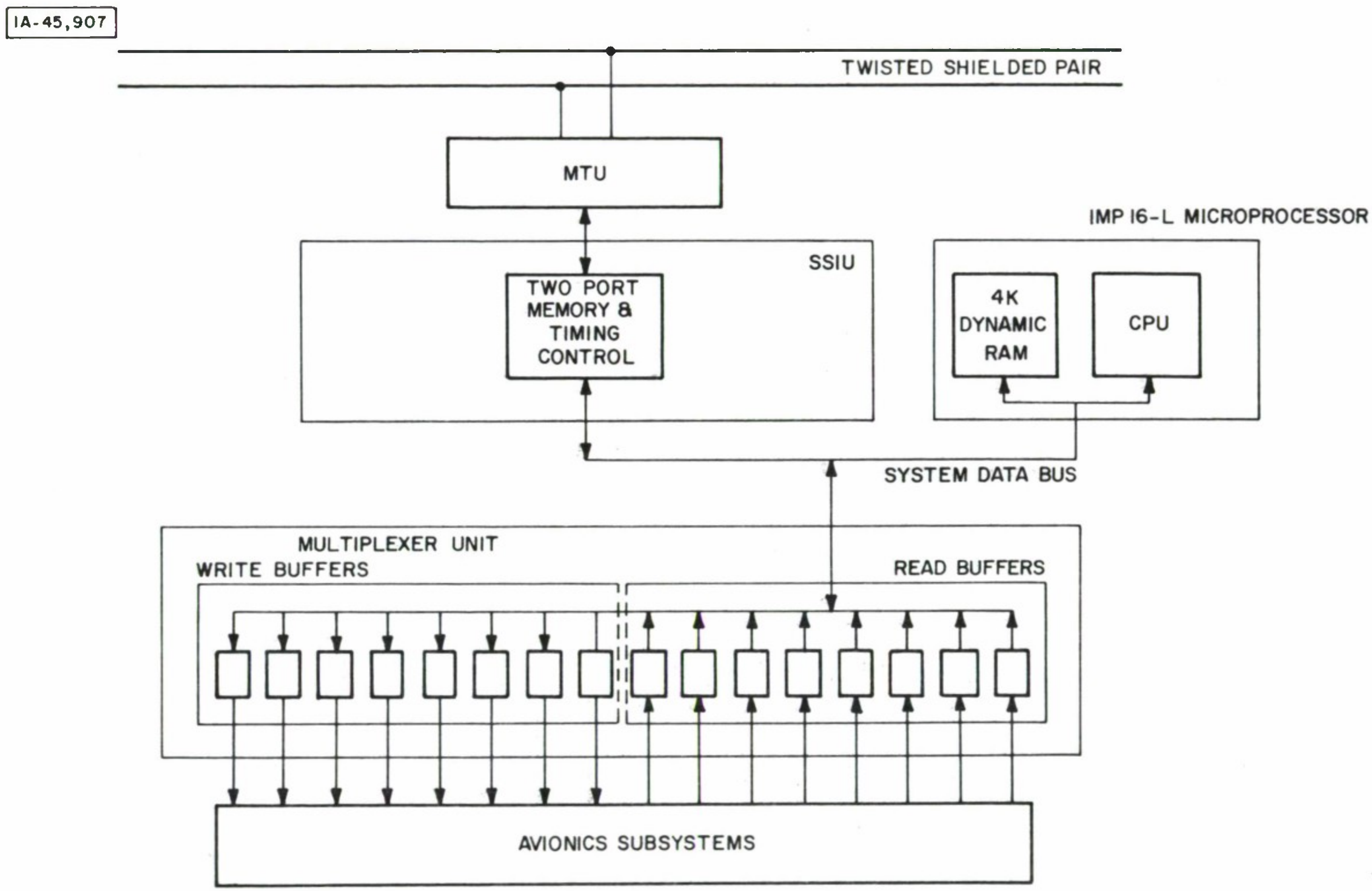


Figure 10 INTERFACE OF THE MICROPROCESSOR/MULTIPLEXER WITH THE REMOTE TERMINAL

through the "B" memory port. The multiplexer system bus is a continuation of the IMP-16L's data system bus and ties the SSIU, IMP-16L and I/O registers together. Both data and address information are multiplexed onto this 16-bit parallel bi-directional bus. Timing and control signals defining data on the bus are sent with the parallel bus lines to each device. The timing control for this bus resides in the IMP-16L. The data on the bus is available to each device, but address decoding resides in control logic at the SSIU's "B" Port and at the I/O register timing and control logic.

In this system each SSIU memory location and each read register or write register appears to the microprocessor as a unique memory location. These memory locations lie within the 64K addressing constraint of the IMP-16L but outside of the 4K of dynamic memory resident in the microprocessor. Table I shows the memory location corresponding to each SSIU location and each input/output register. To read information into or extract information from the SSIU or I/O registers, the microprocessor must execute a standard "load" or "store" instruction.

SSIU/Multiplexer Interface. At the interface of the microprocessor data bus and control lines with the SSIU's "B" memory port, special control and timing logic was developed to synchronize the two processes. In the SSIU the two port memory provides access to the "B" Port, or multiplexer interface, 500 nanoseconds out of each microsecond. When an SSIU read or write memory request is made, the specified memory address and data to be read or stored must be available at the leading edge of the access interval to the memory. The pulse requesting the read or write operation must also be synchronized with the leading edge of the access interval. As mentioned previously, the address and data on the multiplexer data bus appear on the same lines, but are multiplexed in time. The task of the SSIU/Multiplexer Interface is to temporarily steer the address and data information into appropriate buffers and then execute the desired memory operation at the proper time. In reading information from the SSIU memory, an additional timing problem occurs because the SSIU's memory is slower than the IMP-16L's dynamic memory. Under these conditions, the SSIU Interface logic must provide a bus hold condition request until the information from the desired memory location has been retrieved and is placed in an output buffer ready to be read by the microprocessor. Because of synchronization requirements, the SSIU store and fetch operations can hold the multiplex bus for up to 1.5 microseconds.

I/O Register Interface. The interface logic to enable the microprocessor to communicate with the I/O registers is much simpler in design than the SSIU/Multiplexer Interface. This simplicity occurs because data can be read into or retrieved from the I/O registers at the data rates required by the microprocessor system

TABLE I

Microprocessor/Multiplex Address Assignments*

SSIU	SUBADDRESS #1	#2	#3	#4
Word 1	IF00	IF10	IF20	IF30
Word 2	IF01	IF11	IF21	IF31
Word 3	IF02	IF12	IF22	IF32
Word 4	IF03	IF13	IF23	IF33
Word 5	IF04	IF14	IF24	IF34
Word 6	IF05	IF15	IF25	IF35
Word 7	IF06	IF16	IF26	IF36
Word 8	IF07	IF17	IF27	IF37
Word 9	IF08	IF18	IF28	IF38
Word 10	IF09	IF19	IF29	IF39
Word 11	IF0A	IF1A	IF2A	IF3A
Word 12	IF0B	IF1B	IF2B	IF3B
Word 13	IF0C	IF1C	IF2C	IF3C
Word 14	IF0D	IF1D	IF2D	IF3D
Word 15	IF0E	IF1E	IF2E	IF3E
Word 16	IF0F	If1F	IF2F	IF3F

Output Registers
2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007

Input Registers
2008, 2009, 200A, 200B, 200C, 200D, 200E, 200F

* All addresses are hexidecimal.

without timing synchronization or intermediate buffering. The read and write buffers are identical in design to those described in the section of this report dealing with the hardwired multiplexer. Like the hardwired multiplexer design, the system currently will handle eight input and eight output interfaces of 16-bit parallel data. The basic task of the I/O registers is to hold data from the outside world or hold data from the microprocessor until it is required by the equipment interfacing with the multiplex. To achieve this task the microprocessor addresses the I/O registers as a memory address. If the address corresponding to an I/O register is decoded, the accompanying read or write strobe will enable data to be read into or written from the read/write buffers. Expansion of the I/O register capability above the existing eight read and eight output registers would require extending the address decoding logic on the multiplexer timing and control board.

SECTION III

DEMONSTRATION HARDWARE

General

This section provides a description of the avionics subsystems and associated signal conditioning equipment which connect to the RIDS multiplex data bus system for the purposes of providing a hardware demonstration capability in the laboratory. Most of these subsystems were chosen because they were readily available and were representative. The signal conditioning circuits were designed for laboratory demonstration of the RIDS MUX bus system, and are not intended as recommended circuit designs for any other purpose. However, much was learned that would be useful in designing practical circuits. The configuration, philosophy, and scope of the demonstration are also covered in this section.

Demonstration Philosophy

The following guidelines dictated the selection of avionics subsystem hardware for use in the laboratory demonstration:

1. The equipment had to be readily available

2. The hardware had to be capable of producing or simulating different types of signals which were representative of those actually encountered in aircraft avionics equipment.

3. The demonstration was required to exercise a variety of transfer and control functions.

4. The demonstration was required to be of an operational nature.

There is a wide variety of data that is normally distributed throughout a military aircraft. This includes such things as frequency selection control signals for radios, navigation coordinates, flight recorder data inputs, fuel and fuel pressure information, altitude and air speed data, etc. However, all data which can be readily handled on a MUX bus on board an aircraft can be classified into three basic types, analog, digital, and switch closures. Since the multiplex data bus handles only digital data, any data not already in digital form must be converted before it can be distributed by the MUX bus system. In most cases some form of level conversion is also necessary to render subsystem signals compatible with the MUX bus system. Not only are these signal and level conversions required at the source before the data can be accepted by the bus, but usually the signals must be reconstructed into their original form at the sink in order to drive the equipment for which they were originally intended. Certain types of analog signals require further processing because they are referenced signals. For example, a synchro output is a referenced signal where each of the three analog voltages bears not only a specific time, level, and phase relationship with each of the others, but also with a fixed reference ac voltage. Conversion of this type of signal naturally requires different circuitry than the conversion of simple analog signals which are independent, i.e. not referenced.

Avionics equipment selected for the RIDS MUX bus laboratory demonstration require signal and level conversions which are representative of those which would actually be encountered in an aircraft. These include conversion of:

1. Switch closures to digital and digital to switch closure.

2. Analog to digital and digital to analog.

3. Synchro to digital and digital to synchro.

Besides choosing representative types of signal conversion, it was also necessary to demonstrate the transmission of realistic control functions over the MUX bus system. The following control functions were considered to be fairly representative of those performed in aircraft systems.

1. Data transfer, including encoding and/or decoding at the transmitting, receiving, or processing units.

2. Function control, i.e., various combinations of data that, taken together, cause other functions to be performed. Examples include display generation, fault isolation, alarm signalling, command processing, etc.

3. Data sampling. Different types of data require different sampling rates, and the demonstration should show that different sampling rates can be accommodated by the MUX bus, and that the integrity of the sampled signal can be preserved.

In order to place the demonstration in an operational context, it was desirable to include AN nomenclatured pieces of equipment in the set of subsystems serviced by the RIDS bus. The use of AN equipment had the added benefit of providing insight into specific signal conversion problems, ease of retrofitting, desirability of replacing dedicated control and display equipment with integrated or multifunction units, and actual loading requirements for specific equipments.

The following subsection presents a description of the equipment selected, and provides an overview of the demonstration in operation.

Demonstration Configuration

Figure 11 shows a simplified block diagram of the RIDS MUX bus demonstration hardware configuration as it exists in the laboratory. A Digital Equipment Corporation PDP-9 computer is used as the bus controller and is tied through an interface unit to a Sanders 720 CRT Display. The Bus Controller Interface Unit (BCIU) is the interface between the controller and the shielded twisted pair bus. The bus interconnects the controller terminal and the two Remote Terminals (RT). The data sources include an AN/ARC-50 control unit at RT #1 and two potentiometers and a synchro transmitter at RT #2. These are described in the following paragraphs.

AN/ARC-50

The AN/ARC-50 control unit provides control data such as frequency selection, power settings, mode selection, etc. which is converted to digital information and then clocked into the SSIU multiplexer input registers (write buffers). From there they are transferred into the SSIU memory via the "B Port" (see section on SSIU hardwired multiplexer). The bus controller periodically calls for this information, and it is sent via the SSIU "A Port", the MTU and the bus.

The volume control information from the ARC-50 control unit is decoded by the bus controller and is formatted for display on the Sanders 720.

The frequency selection information is retransmitted by bus controller back over the bus to RT #1 where it is stored in the SSIU memory in the subaddress and word location associated with the frequency indicator. From there it is clocked into the corresponding SSIU multiplexer output register (read buffer) and then through a signal converter to the frequency indicator where it is used to drive the dc resolvers in the unit which cause the proper frequency to be displayed.

The power setting and mode selection information are forwarded by the bus controller over the bus to RT #2 where they are stored in the SSIU memory address associated with the AN/ARC-50 Transmitter/Receiver (T/R) unit. From here they are transferred under control of the IMP-16L microprocessor to the SSIU output registers and then through signal conversion circuits to the ARC-50 where they are used to control the power setting and mode selection functions in the unit.

18-45,917

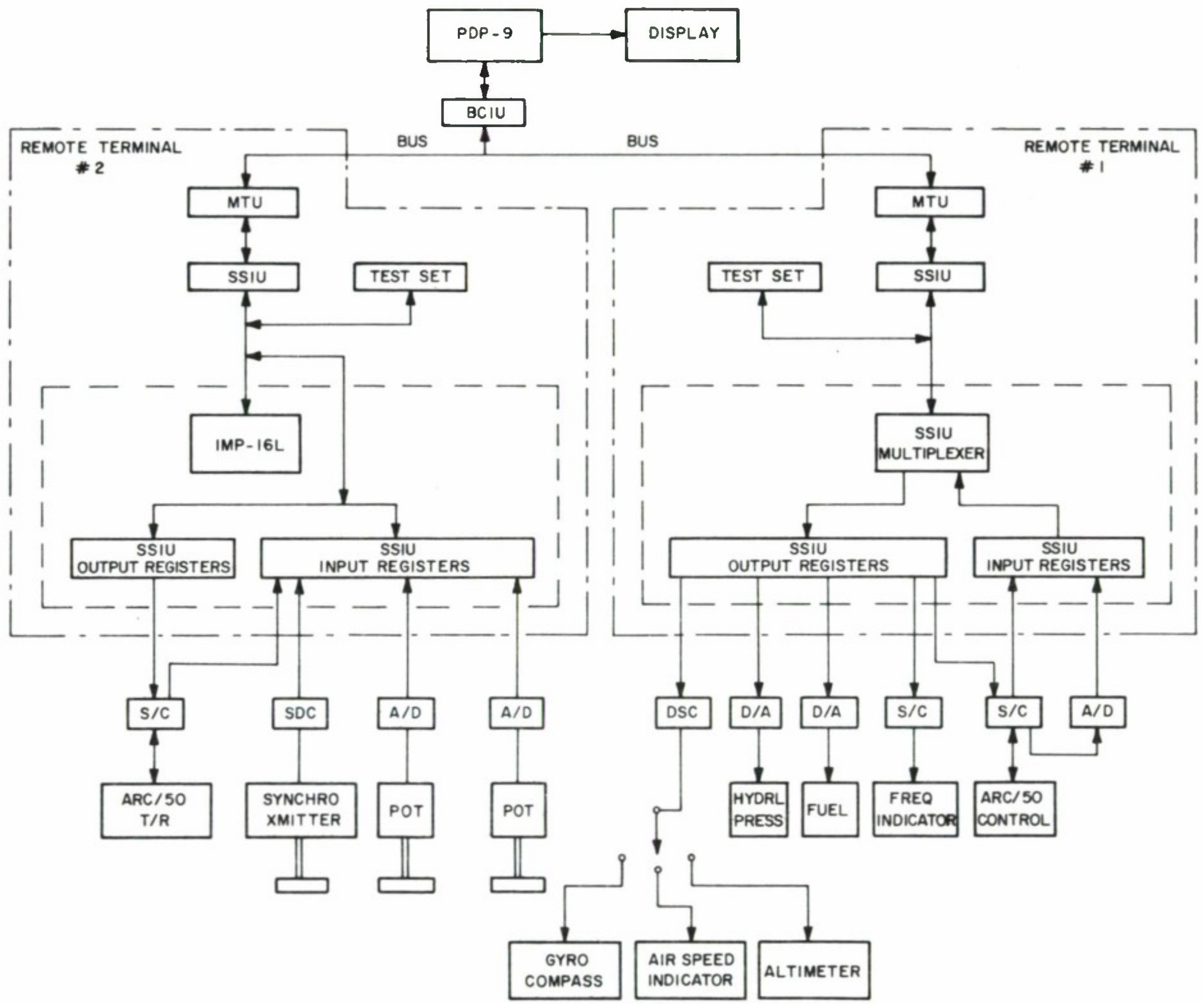


Figure 11 RIDS MUX BUS DEMONSTRATION HARDWARE CONFIGURATION

Fuel and Hydraulic Pressure

The potentiometers at RT #2 are used to simulate transducer outputs from fuel and hydraulic pressure sensors. The analog voltages are digitized by means of A/D converters and loaded into the SSIU input registers and from there into the SSIU memory. The data is requested by the bus controller, and when it is received, it is formatted for display on the Sanders unit. The bus controller also transfers this information (not formatted for display) to RT #1 where the data is reconverted to analog signals by means of D/A converters and used for driving the fuel and hydraulic pressure gauges.

Gyrocompass, Air Speed Indicator, and Altimeter

The synchro data generated at RT #2 is converted to digital data by means of a synchro to digital converter circuit and is then clocked into the SSIU input registers. This information is then sent to the bus controller where it is formatted for display. The data is also transferred to RT #1 where it undergoes digital to synchro conversion. The converted signal is used in conjunction with a 400 Hz reference signal to drive any one of three synchros, depending on the position of a wafer selection switch. The three synchros control the pointers on a gyro compass, an air speed indicator, and an altimeter.

Sampling Rates

A capability for varying the sampling rates is provided for demonstration purposes. Changing one bit in any one of four words by means of switches on the test set front panel at RT #1 causes the controller to transmit data to the RT at varying rates.

Signal Conversion Circuits

Signal conversion circuits were designed for all avionic subsystems used in the RIDS demonstration configuration. The types of signal conversion include dc-to-digital-to-dc (two and three level), analog-to-digital, digital-to-analog, synchro-to-digital, and digital-to-synchro. As was mentioned earlier in this report, avionics subsystems were selected for use in this demonstration based not only on their availability but more importantly on how well they represented typical aircraft avionics equipment. The types of signal conversion required for the demonstration, therefore, are representative of those which would be required in a military aircraft equipped with a multiplex data bus system of this type. The signal conversion allows AN designated avionics equipment of the types presently existing in the USAF inventory to interface and operate with a multiplex digital data bus designed to MIL-STD-1553.

SECTION IV

SYSTEM OPERATION

General

This section discusses the operation of the RIDS multiplex data bus system as it is currently configured. With the exceptions noted earlier in this document, the RIDS bus has been designed in accordance with MIL-STD-1553 (USAF). Figure 12 shows a simple block diagram of this system.

Word Formats

The system operates in a command response mode in which all messages are initiated by the bus controller and responded to by the remote terminals. There are three basic word formats used for the transmission of information throughout the system. These are illustrated in Figure 13, and they are (a) the command word, (b) the data word, and (c) the status word.

Command Word

A message transfer is always initiated by a command word from the controller. This word is headed by a command sync pulse (generated in the BCIU before the word goes out on the bus), and it includes a terminal address which alerts the MTU in the particular remote terminal to which the message is addressed, a transmit/receive bit which notifies the remote terminal to prepare to receive data or to transmit data, a subaddress which instructs the RT which SSIU memory location is to be used for storage or retrieval of the message data, a data word count which specifies the length of the message, and finally a parity bit (generated in the BCIU) which provides a means for checking the validity of the word at the receiving end.

Data Word

The data word (Figure 13b) is headed up by a sync pulse which is the inverse of the command pulse. The sync pulses are three bit times (i.e. 3 microseconds) in length. Following the sync pulse in the data word are sixteen information bits and one parity bit. The total word length is equivalent to twenty bits (20 microseconds).

Status Word

The status word (Figure 13c) is always generated in the remote terminal involved in the message transfer, and it is always the

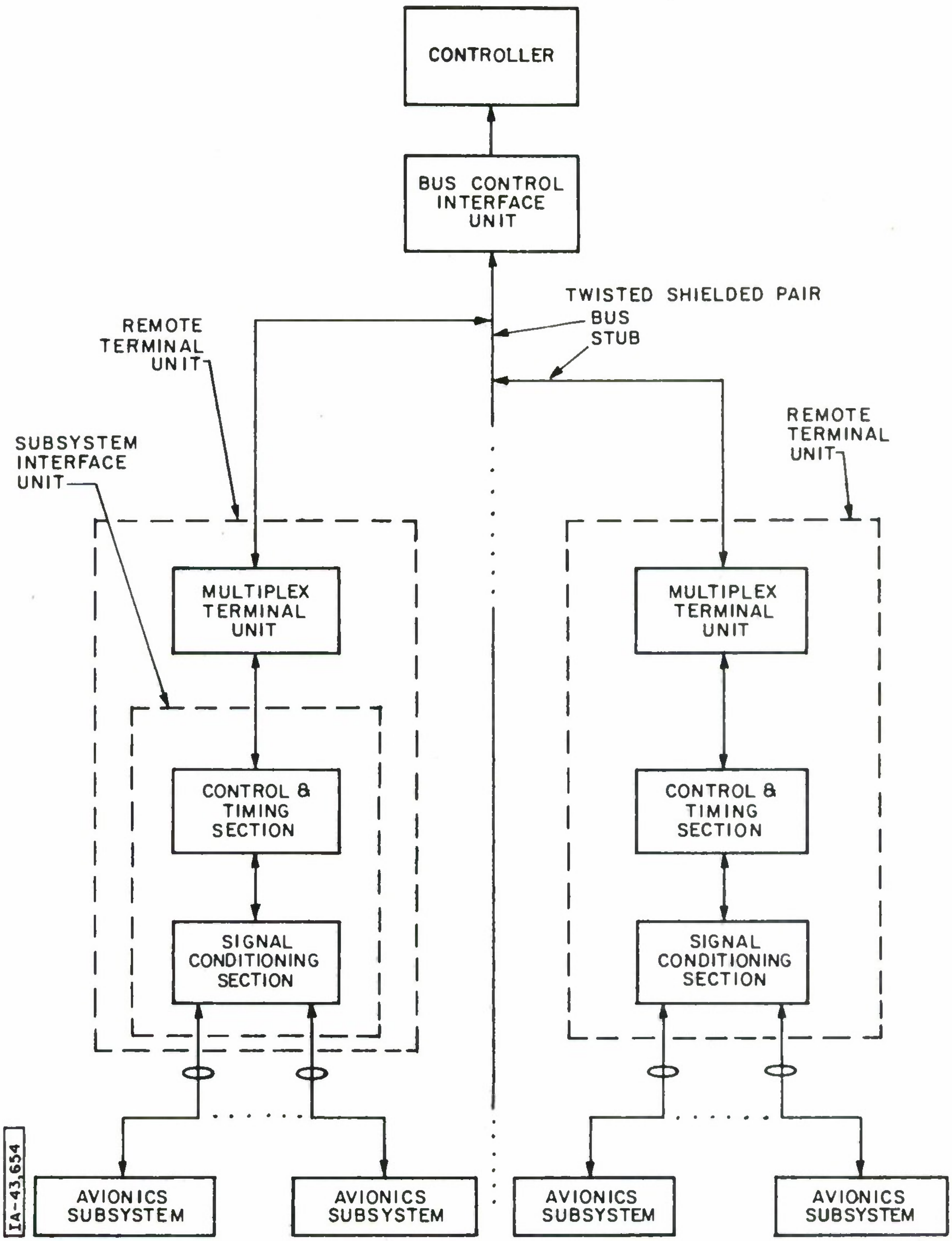


Figure 12. A "STANDARD" MULTIPLEX BUS
MIL-STD.-1553 (USAF)

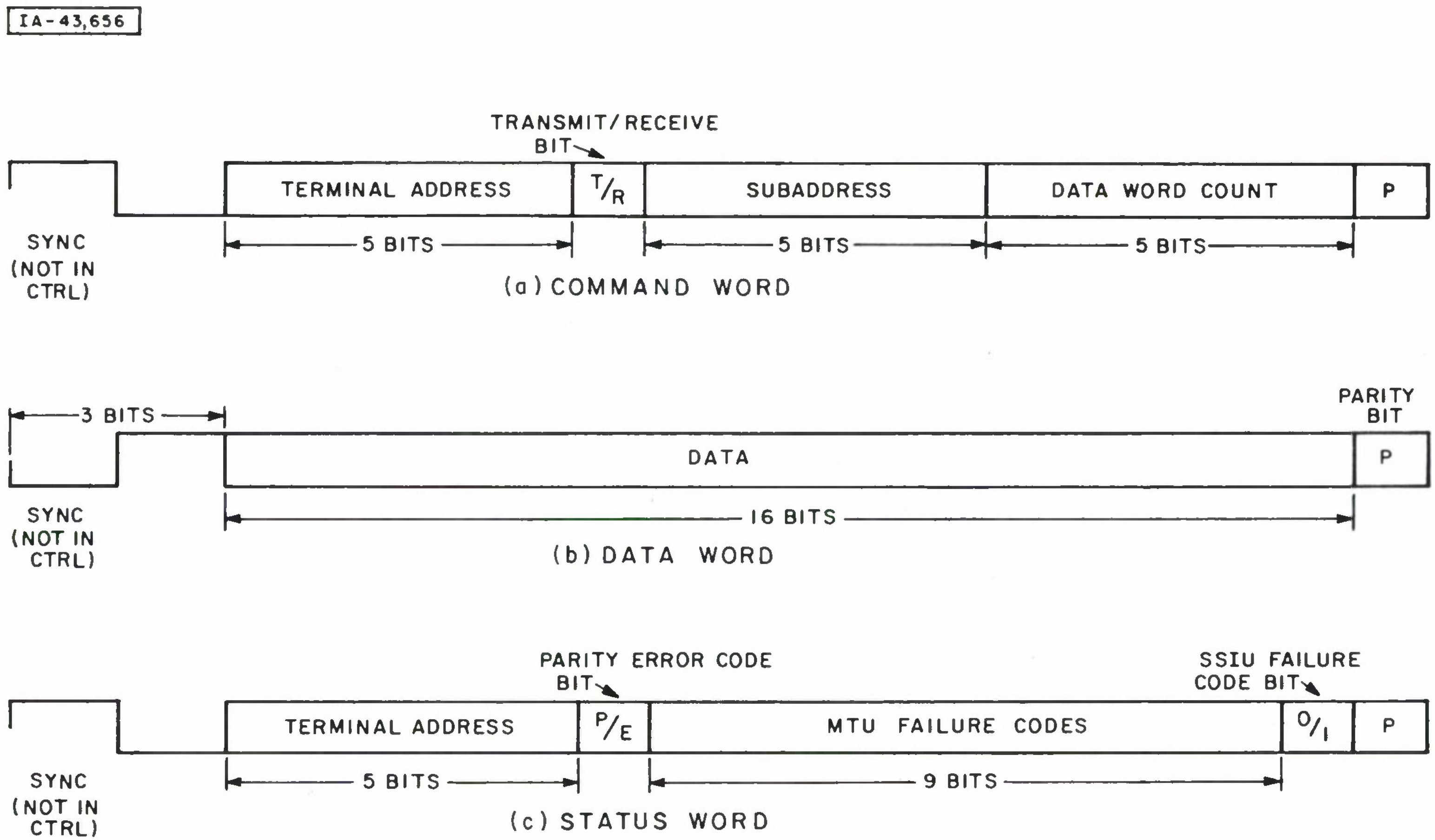


Figure 13. WORD FORMATS MIL-STD-1553 (USAF)

first word transmitted by the responding RT. If the RT is receiving data from the bus, the status word is transmitted after the receipt of the last data word; if the RT has been commanded to transmit data onto the bus, the status word precedes the data words. The status word is headed by a three microsecond sync pulse identical to the command sync. The only difference is that the command sync is always generated in the BCIU and the status sync is always generated in the MTU. Following the sync pulse are five bits which represent the sending terminals address, a parity error bit which designates whether or not a parity error was detected in any of the message words received by the RT, nine bits used to indicate various MTU failures (not used in the present laboratory demonstration system), an SSIU failure code bit, and finally a parity bit. The controller takes various actions on the basis of the status word.

Message Formats

There are three message formats used in the transfer of information over the RIDS multiplex data bus system. These are illustrated in Figure 14, and are (a) terminal to controller, (b) controller to terminal, and (c) terminal to terminal.

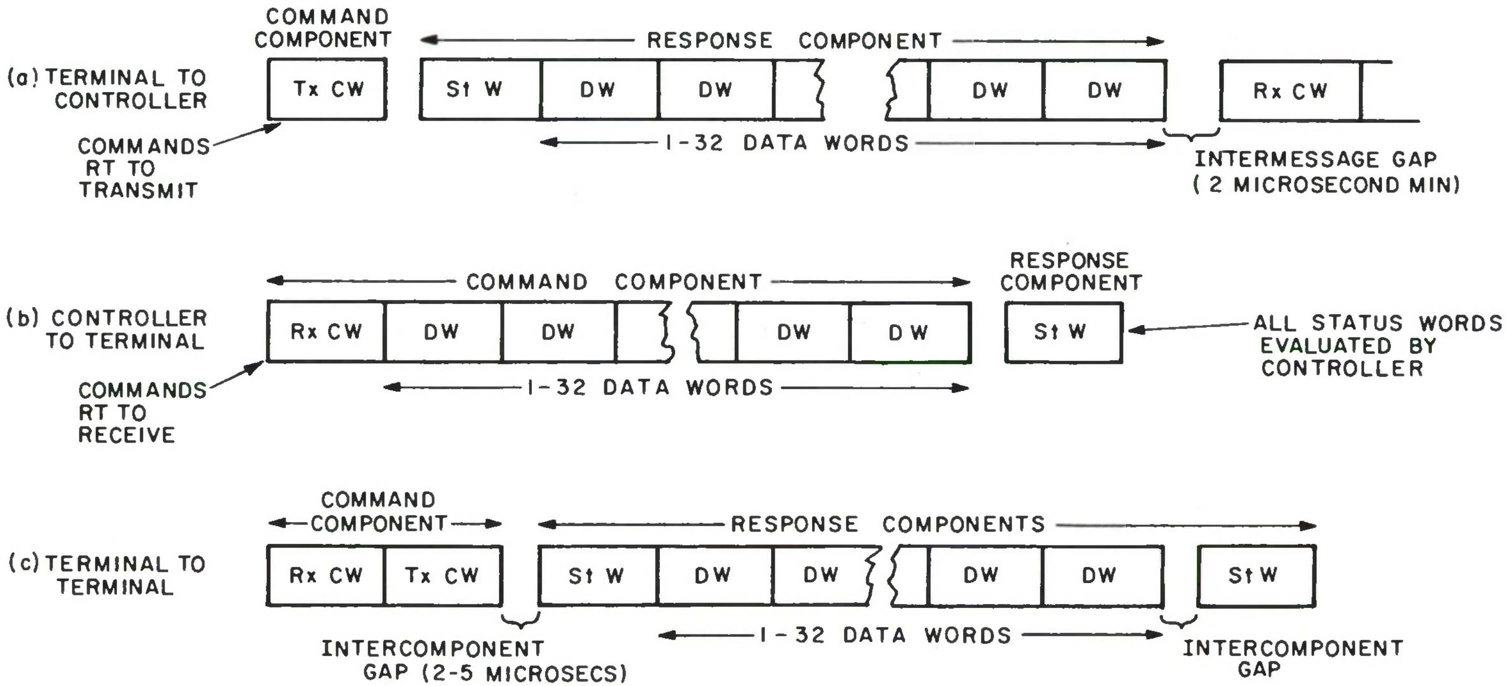
Terminal to Controller

In the terminal to controller mode of operation the message is initiated by the controller which sends a transmit command addressed to a particular remote terminal. The addressed RT responds to the command by sending out a status word followed by the number of data words called for in the command word (up to a maximum of 16 in the lab demonstration configuration; the standard calls for a maximum capability of 32). The intercomponent gap (i.e., the dead time on the bus between the last bit in the command component and the first bit in the response component) is between 2 and 5 microseconds in duration. The command component is defined as the sequence of contiguous command and data words transferred from the controller. The response component is defined as the sequence of contiguous words transferred by the remote terminal in response to a command from the controller.

Controller to Terminal

In the controller to terminal mode of operation the message is again initiated by the controller. The command word from the controller is a receiver command which activates the receiver circuits in the MTU of the remote terminal which was addressed by the command word. As can be seen in Figure 14(b), there is no gap between the command word and the first data word since both words are a part of the command component transmitted by the controller. The number of data words (up to 32 in the standard and 16 in the

IA-43,655



(1) COMMAND COMPONENT: SEQUENCE OF CONTIGUOUS WORDS TRANFERRED FROM CONTROLLER TO REMOTE TERMINAL.

(2) RESPONSE COMPONENT: SEQUENCE OF CONTIGUOUS WORDS TRANSFERRED FROM RT TO CTRL OR RT TO RT.

Figure 14 MESSAGE FORMATS ON MULTIPLEX BUS demonstration system) in the message again corresponds to the data word count code in the command word. Within 2 to 5 microseconds after receipt of the last data word in the message, the RT responds with a status word.

Terminal to Terminal

Figure 14(c) illustrates the terminal to terminal message format. Here again, the message is initiated by the controller. In this case, however, it is necessary for the controller to transmit two command words since two RT´s are involved. The first word is a receive command addressed to the RT which is to receive the message. This command places that RT in the receive mode of operation, and it will remain in this mode until it has received the total number of data words indicated in the command word. The second command word transmitted by the controller is addressed to the RT which is to transmit the message. Within 2 to 5 microseconds after receipt of this command word, the second RT transmits a status word plus the commanded number of data words. The status word is ignored by the first RT, but it is monitored and evaluated by the controller. The data words, however, are accepted by the first RT, and 2 to 5 microseconds after the receipt of the last data word, the RT transmits its status word which is also received and evaluated by the controller.

System Operation

As stated in previous sections, all messages transferred over the RIDS multiplex digital data bus are initiated by the bus controller which initiates action through its software program. The controller loads its I/O register with information relative to the number of data words to be transferred in the message and the location where they reside in memory (the address of the first data word in the message is given). The channel-to-channel interface then assumes control and issues an "initiate write" command which notifies the BCIU that data is ready to be transferred. The BCIU responds with a "word ready" pulse. Upon receipt of this pulse, the bus controller places the first word on the 16-bit parallel I/O bus and simultaneously issues a "write strobe". The first word is the PRELUDE WORD, and after transfer it is stored in a BCIU input buffer and decoded to determine the mode of operation requested by the controller. The BCIU sends the controller a "word ready" pulse after receiving and decoding the PRELUDE WORD, and the controller responds by placing the COMMAND WORD on the I/O bus. At this point the controller interface section of the BCIU initiates action to transfer the COMMAND WORD to the bus interface section. This action consists of raising or lowering the "T/R" line depending on whether the transmit mode or the receive mode of operation is being called for by the controller. The data in the command word buffer is placed on the command word lines, and a "command load" signal is given. The command word is then transferred into the bus interface section of the BCIU. In the controller to terminal mode of operation, the bus controller then gates each data word of the message in sequence onto the data lines, and they are immediately transferred to the bus interface section of the BCIU and stored in its memory. When all the data words of the message have been transferred in this way and stored in memory, the bus controller signifies the end of message, and the BCIU commences to serially clock the command word followed by the data words onto the bus at a 1 megabit per second rate. All information is encoded in Manchester II before being placed on the bus.

Both remote terminals receive the sync pulse of the command word, and their receive circuits check it for validity. If both halves of the pulse are of the proper width and phase, the command circuits in the MTU's are enabled, and the next five bits in the command word (the address code) are compared with their own assigned address code. The RT in which the address compare checks positive accepts the remainder of the message. The RT in which the address check is negative disregards the message. The remainder of the command word is decoded in the receiving RT, and the T/R bit sets the MTU in the transmit mode if it is a logic "1", and into the receive mode if it is a logic "0". The word count information in the command word sets a word count register which keeps track of the number of data words in the message.

In the receiving mode, the data words are received serially from the bus and stored a word at a time in parallel in the MTU memory. When the last word has been received, the MTU starts serially clocking out a status word back to the controller over the bus while at the same time it transfers the command word and all of the data words in sequence over a 16 parallel line data bus to the SSIU. The SSIU decodes the subaddress and word count in the command word and stores the data words in its memory beginning at the first location of the subaddress and continuing sequentially in consecutive address locations until the proper number of data words have been stored.

In the transmitting mode, the MTU transfers the command word to the SSIU immediately, and at the same time starts clocking out the status word onto the bus serially. The SSIU transfers from its memory the number of data words called for in the command word over the 16 parallel line MTU/SSIU interface. The starting location in the SSIU memory from which the words are taken is the first location of the subaddress given in the command word. The data words are transferred over the MTU/SSIU interface in 16-bit parallel words at a 1 MHz rate. Up to 16 data words are transferred to the MTU and stored in its memory during the 20 microseconds required to clock the status word onto the bus. As soon as the last bit of the status word is transmitted, the first data word is parallel loaded into the I/O shift register. This word is then serially clocked through the transmit circuits which first generate a data sync pulse header and then convert the bits into Manchester II code before transmitting them onto the bus. The word count register keeps track of the number of words sent, and when the number equals the number called for in the command word, the transmit circuits are disabled and the RT is ready to receive the next command word.

On receiving data from the bus, the MTU checks every word to ensure that it has the proper odd parity. If any word in the message has even parity, an alarm circuit is activated, the message is not stored in the SSIU, and the C/E bit in the status word returned to the controller is set to logic "1".

On transmitting data onto the bus, the MTU generates the proper parity bit and adds it to each word.

The BCIU receives data in much the same way that the MTU does. After the words have been checked and put into memory, a signal is sent to the computer notifying it that the first word is on the line and ready to be transferred. The computer, which has been waiting in a read loop since it sent out the last message, now accepts the data from the BCIU through the channel-to-channel interface.

The avionic subsystems access the SSIU memory through its "B" port as described in an earlier section of this report, whereas the MTU accesses the SSIU memory through its "A" port. The "A" port access is accomplished only during the positive half cycles of the 1 MHz clock, and the "B" port access takes place during the negative half cycles. This precludes interference between the MTU and the subsystems. The subsystems can therefore write data into the SSIU memory or read data out of the memory at their own sampling rates without regard for MTU interaction with the SSIU.

Software for the Laboratory Demonstration

When designing a software package that is intended to perform a number of different functions, it is expedient to partition the programs so that their interdependence is minimal. For this reason, the message handling software developed for the control of the multiplex bus was structured to permit the transfer of information between the subsystems serviced by the bus essentially independently of the nature of those subsystems and the specific content of the data being transferred. The interface between the message control programs and the application dependent processing was confined to a single entry/exit routine, which permitted processing of the data word content to any degree of complexity without necessitating modification of the basic control programs. As a consequence of this, the additional programming required to service the AN/ARC-50 UHF radio and the miscellaneous cockpit displays, e.g., instrument gauges, warning displays, etc., was primarily in the area of applications processing.

The following subsections outline the programs that were generated to interpret and display the content of the data words being transferred between the subsystems, via the bus. In addition, to permit the reader to appreciate the relationship between the application and control software, a brief description is given of the basic message control cycle.

Outline of Message Control Software

The software package developed to control the experimental bus served two distinct purposes. One of these was the message handling function itself; the other consisted of a number of programs which permitted the majority of the message handling routines to be debugged without recourse to the hardware they were intended to control. To the level of complexity implemented in the experimental bus work, this test/simulation software has been extremely useful. It enabled the basic message cycle to be completely debugged prior to interfacing the control programs with the bus hardware, and subsequently allowed the bus controller to act as a sophisticated signal generator for the debugging of the bus subsystems in the later stages of system integration. The main functions implemented in the message handling software are listed in Tables IIa and IIb.

For convenience of conceptual design, together with the desirability of developing a modular computer program, the message control function--a subset of the overall bus control function--was subdivided into a number of discrete operations:

- Message discrimination
- Message synchronization
- Command/Response component transfer to/from the control unit to the bus
- Message validation
- Common response processing

TABLE IIa

Control Software Functions: Operational

- Initiates and controls three modes of message transfer

    a) Controller to Remote Terminal

    b) Remote Terminal to Controller

    c) Remote Terminal to Remote Terminal

- Samples subsystems serviced by the remote terminals at one of six rates: 32, 16, 8, 4, 2, 1 per unit time period.

- Sequences messages to obtain a quasi-uniform loading of the multiplex bus throughout each unit time period.

- Checks validity of status words associated with each response component, and prints out address of subsystem location if an error is indicated.

- Distributes data words contained in incoming response components to core locations accessible to applications software.

TABLE IIb

Control Software Functions: Test and Development

In the course of developing the MUX bus control software, the following options within the basic message processing cycle were included primarily for the purpose of debugging and testing.

- Option 1: Include/exclude all Command Component transfers

- Option 2: Include/exclude Command Component transfers to Bus Controller Interface Unit

- Option 3: Include/exclude Command Component transfers to internal print buffer

- Option 4: Option 3 can be changed/not changed without recycling program

- Option 5: Include/exclude all Response Component transfers

- Option 6: Response Component from BCIU/Response Component from response simulation in internal buffer

- Option 7: Minor cycle initiation under clock control/no clock control

- Option 8: Include/exclude "no response" processing

These activities must be repeated for each command/response message sequence; the latter consisting of a command component generated by the bus controller, and its associated response component originating from an MTU(s). Thus, if the bus load consists of n command/response sequences per second, the basic set of message handling operations must also be executed n times per second.

The message handling cycle is shown diagrammatically in Figure 15. Two additional operations are included which do not constitute part of the "per message" activities listed above. One of these--sequence initiation--is required for entry into the repetitive message handling loop when the bus system is switched on. The other--applications, or special response processing--consists of the routines dependent on the content of the data words, and comprises the bulk of the software generated for the present demonstration.

The figures given in parentheses--under the blocks in Figure 15--refer to the number of memory cycles and machine language instructions, respectively, used to program the routines on a PDP-9 computer (memory cycle time one microsecond). The reader is referred to Reference 3 for a discussion of the significance of these numbers.

The routine that is of more immediate interest is the common response processing, since it contains the interface between the message handling cycle and the applications processing used to interpret the content of the data words. The position of the data words within a command/response message sequence is established by the bus protocol defined in MIL-STD-1553 (USAF). In the particular case of a remote terminal to controller message sequence, a single command word is transferred to a given MTU--determined by the appropriate address field. The MTU responds with a status word and a number of data words--determined by the content of the word count field in the command word. In the bus controller the incoming response component is placed in a data buffer, and each word is then processed according to a predetermined sequence.

The first word--the status word--is checked for validity, and if determined to be in error, a message is printed on the TTY associated with the controller, giving the terminal and subaddress from which the response component was received. The validation criteria are common to all status words, and the latter are handled similarly for all messages.

The remaining words in the response component are data words which, in general, require unique handling dependent on the subsystems at which they originated. The calling of, and return from, the appropriate application routine(s) required for each data

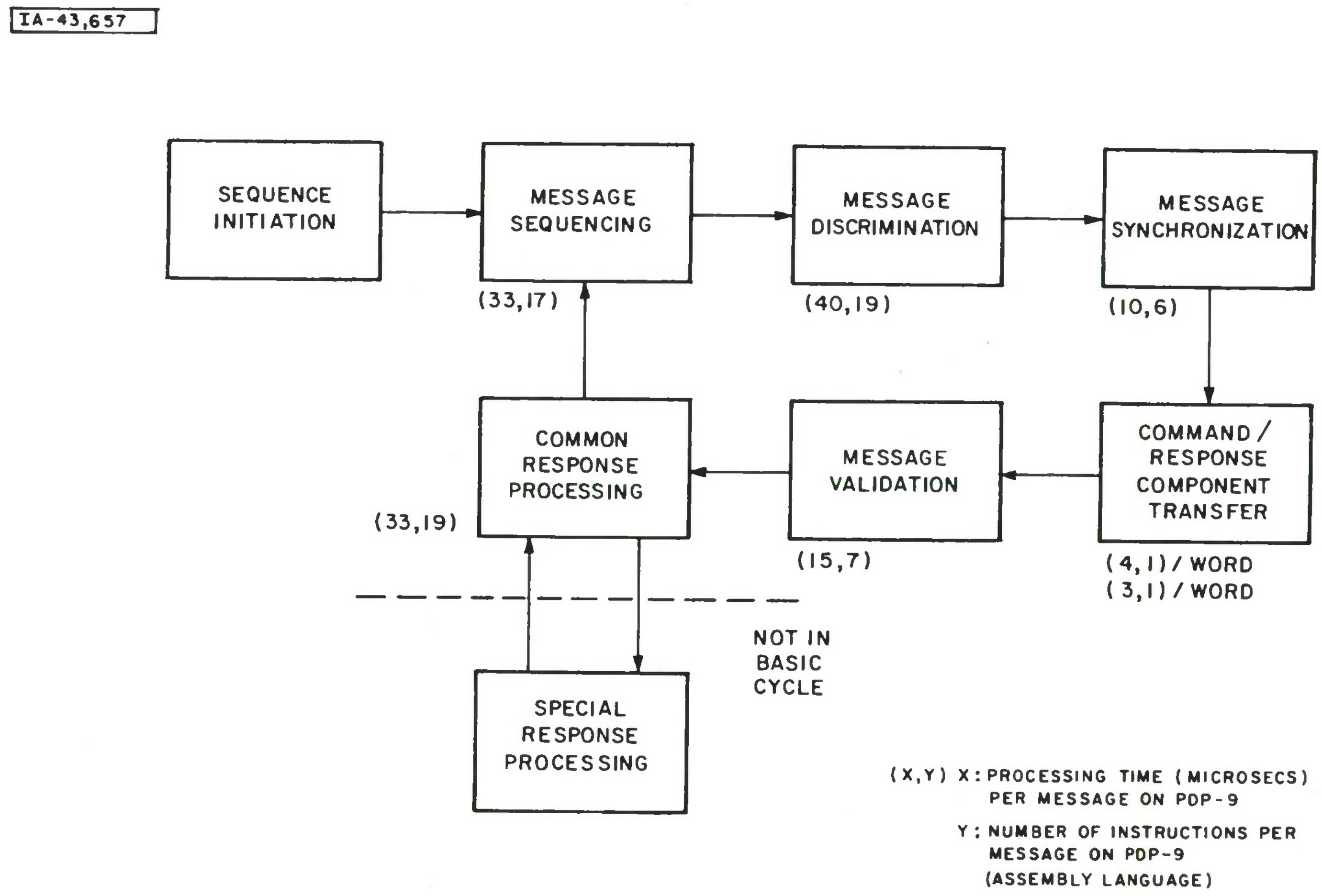


Figure 15. EXPERIMENTAL BUS CONTROL SOFTWARE : BASIC CYCLE

word constitutes the processing interface between the conceptually distinct functions of message transfer and message usage. The flexibility of processing necessary to cope with the different types of data word content, e.g., navigational data, flight control information, etc., is obtained by means of a system of pointers which is best described diagrammatically, see Figure 16. Each command/response message sequence that has data words in the response component has a processing word stored contiguously with the message's command word in core. The processing word points to a table of pointers--with an entry for each data word in the response component. Each of the pointers initiates the applications processing required by its associated data word.

As has been mentioned previously, the basic message control software, and in particular, the common response processing, was structured to permit message handling independent of the types of equipment being serviced by the bus. Thus, the task of generating additional software resulting from the selection of particular equipments, e.g., AN/ARC-50, gyro heading repeater, etc., for demonstration, resolved itself into one of developing applications software--as distinct from bus control programs. The following subsections outline the multiplex bus capabilities that have been demonstrated, the message flow that was required to implement the demonstration, and the applications software that was developed to utilize the data words transferred.

Multiplex Bus Capabilities Being Demonstrated

The demonstration of the capabilities of the multiplex bus falls into two fairly distinct phases. The first of these shows how elementary operations, such as data transfer, function control, etc., can be implemented on the bus. The method of displaying that these activities are indeed taking place, is by panel lights that monitor the contents of the two port memory associated with the subsystem interface unit. The second part of the demonstration is more operationally oriented. It includes the remote control of an AN/ARC-50 UHF radio, remote display of various aircraft parameters on standard cockpit indicators, threshold monitoring with diagnostic/warning display of critical conditions, etc.

Data Transfer. The purpose is to show the transfer of a data word between SSIU locations, via the bus controller.

The controller interrogates, at a predetermined rate, a given location in an SSIU's memory--defined by an MTU address, subaddress and word count--and transfers the content to some other location, similarly defined. The transfer is via the controller, and uses a terminal/controller message sequence followed by a controller/terminal transfer. The "applications" processing of the

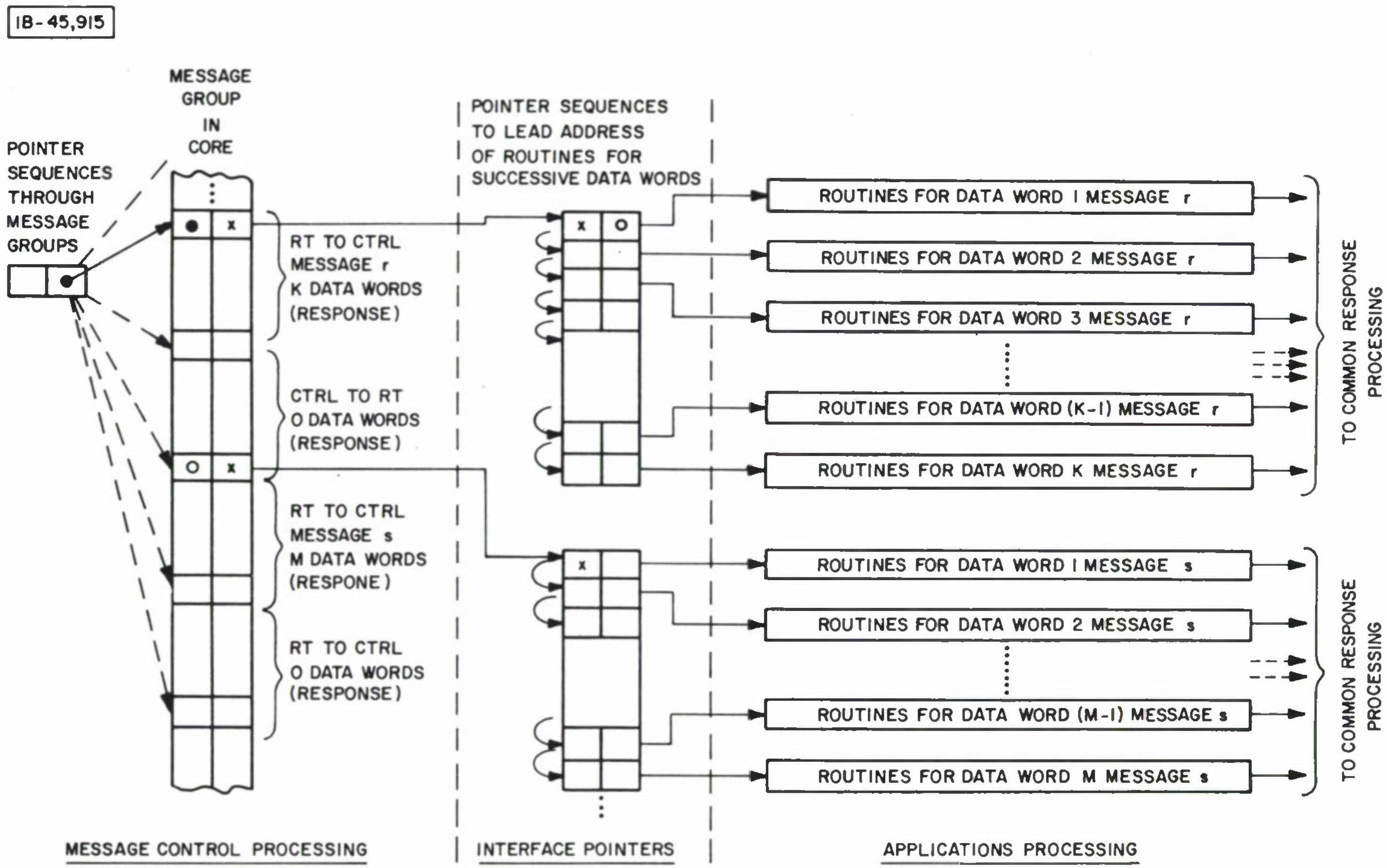


Figure 16 INTERFACE BETWEEN MESSAGE CONTROL AND APPLICATIONS PROCESSING

data word is purely that of translation; from the response buffer in which the incoming data word is stored, to a data word location in an outgoing message.

Function Control. The purpose is to exhibit the remote initiation of a function with a single discrete.

The bus controller interrogates--at a predetermined rate--a given location in an SSIU's memory. The data word is tested for the condition of a "switch" bit. When clear, (0), a data word containing all zeros, is transferred to another SSIU location. When the "switch" bit is set, (1), a data word containing 101010 and 010101 on alternate transmissions, is placed in that location. Thus, the effect is to remotely activate a function, as can be seen by the flashing lights of the test panel when it is set to monitor the second memory location.

The message handling processing in the bus controller consists of first initiating a terminal/controller and then a controller/terminal message sequence. The application processing involves interrogation of the incoming data word to determine the condition of the switch bit; conditional branching to alternate routines to set the appropriate content of a core location to perform the function; followed by transferring the contents to a predetermined data word in a message addressed to the given location in the SSIU's memory.

Signal Sampling Rates. The purpose is to demonstrate the different rates at which data can be sampled from a subsystem, and transferred between subsystems.

The bus controller interrogates a data bit at a given SSIU location. When the bit is set, a data word containing fifteen zeros and a one is transferred to some other SSIU location. Each time the data bit is interrogated, i.e. at the sampling rate, the one bit in the outgoing word is shifted one place to the left. After unit time has elapsed, the number of positions through which the bit has been stepped corresponds to the sampling rate.

The above procedure is repeated for four source/sink pairs operating at different sampling rates--1, 2, 4, and 8 samples per unit time. The net effect is a set of stepping panel lights, each incrementing at a rate at which the particular source is being sampled and transferred to the sink location.

Operation of the AN/ARC-50 UHF Radio. The purpose of the AN/ARC-50 demonstration is to remotely control the radio, e.g., frequency selection, operating mode, transmitter power, etc., and to display its parameters by panel indicator and CRT.

The data transfer, by means of terminal/controller and controller/terminal message sequences, is identical in form, although different in content, to that used in the demonstrations outlined above. However, the applications processing is considerably more extensive. The additional complexity arises from the need to extract eight parameters, describing the condition of the radio, from the incoming data words. In general any given parameter can take one of several states, e.g., the frequency setting can range between 200.00 MHz and 399.95 MHz in 0.05 MHz increments. At the operator's option, the parameters can be displayed in tabular format on a CRT. The rate at which the radio's controls are sampled is compatible with the response time of the radio control unit and the T/R unit combination. The transfer of the control signals via the multiplex bus causes negligible degradation in response time of the control mechanism.

Remote Display of Aircraft Parameters on Cockpit Indicators. The purpose of the demonstration is to show how flight and aircraft parameters, e.g., gyro compass heading, fuel remaining, etc., can be sensed at various locations and displayed on standard cockpit indicators, and in summary format on a CRT, using the multiplex bus as the transfer medium.

In terms of the data processing involved, the message handling routines are the same as those used for the other demonstrations outlined above. The applications processing was similar in form, but different in detail.

Monitoring and Diagnostic Functions. The purpose of the demonstration is to show that monitoring and diagnostic functions can be implemented using the multiplex bus.

When the data transfer between subsystems is via the bus controller, it is an obvious extension from the control and display of subsystem parameters to their monitoring and fault diagnosis based on their condition. Using the "fuel remaining" and "hydraulic pressure" parameters, threshold tests on their levels are made, and flashing DANGER symbols are superimposed on the summary CRT display when the values fall below predetermined levels. In addition, another warning is activated by the same stimulus and made auditory by means of a warning horn.

The function of fault diagnosis is also demonstrated using the operator options for selection of the CRT display format. If, inadvertently, the switch settings are set for two displays simultaneously, a diagnostic display is automatically called up that informs the operator that an illegal switch setting has been made.

Another implementation of fault monitoring is based on the content of the status word(s) that form part of each command/response message sequence. Each status word is checked by the bus controller. If the error bit is set, the address and subaddress of the terminal are printed on the TTY.

Outline of "Response Handling" Software for the Laboratory Demonstration

Detailed discussion of specific programs may be rather tedious. Therefore, attention will be confined to the operational type demonstration software at a functional level. Details of the coding will not be considered.

The processing used on all incoming messages is divided into two distinct types. One of these is common to all response components of the command/response protocol, and is concerned with routing the incoming data words to the appropriate applications routines. The second part of the response handling is the application processing which may vary from word to word, dependent on the content, e.g., navigational data, flight control data, etc.

If the requirement is for the data word to be deposited in some specific location in core, then a simple procedure for this purpose is activated. If, on the other hand, the data word is packed with several signals destined for a number of different users, and different applications processing is required on each, then the initiation procedures are more complex. In all cases, whether single or multiple signals and users are involved, the processing flow is controlled by a sequence of pointers, shown diagrammatically in Figure 16. The pointer which forms the base of this "tree" is the sole point of contact between the bus control processing and that part of the applications processing being done in the computer which is handling the bus control function. In terms of the message handling software the flexibility necessary to permit the bus system to service a wide range of subsystems resides in, and is confined to, the series of pointers which initiate the various applications routines required by the data word content.

As a slight digression, it should be noted that even when the information content of the individual data words is different, e.g., one carries range and another bearing data, there are frequently similar operations to perform on each; for example, scaling, BCD conversion, etc. Thus, it is possible to incorporate a considerable degree of commonality into the applications programs at a structural level below that of the bus control software.

It has already been mentioned that the invariant portion of the software handling the response components had already been developed, see Reference 3, prior to the request for the present demonstration. In consequence, the additional work required for the demonstration was confined to the development of the applications programs and their interfacing with the message control software. The latter was purely procedural. It consisted only of setting up tables of pointers. The applications processing was specific to the type of subsystem being serviced, e.g., the UHF radio and the cockpit indicators, and is the subject of the descriptive material given below.

Outline of Applications Software for AN/ARC-50 Demonstration

The use of a multiplex bus system for the transfer of information between subsystems or separated units of a subsystem is conceptually straightforward. In practice, the constraints imposed by the desirable goals of standardization and future flexibility may result in a message flow between units that appears on the surface somewhat redundant. This will be pointed out in the following discussions.

Information Flow. The information flow between the three units comprising the AN/ARC-50 when dedicated wiring is used is shown in Figure 17a. In terms of a centralized bus control network, this information exchange translates into the word flow shown in Figure 17b. In the context of the MIL-STD-1553 (USAF) bus protocol, this evolves into the message interchange shown in Figure 17c. It will be recalled that the standard specifies the use of a command/response discipline; consequently, each information transfer requires a message sequence consisting of a command component and a response component, irrespective of whether the data words are moved from terminal to controller or vice versa. Each of these transfers incurs an overhead of one command word and one status word if the data word transfer is to or from the controller. The overhead is two command and two status words if the terminal to terminal mode of operation is used. For the AN/ARC-50 demonstration, terminal/controller and controller/terminal transfers were used, since the information being moved was required at the controller for purposes of interpretation and display.

It should be stressed that the implementation of any given information transfer under the constraints of MIL-STD-1553 is not, in general, unique, and there is flexibility to adapt the number of message sequences used to conform with other conditions. In this context it can be seen that there is a marked difference in the number of data words required to operate the AN/ARC-50 as indicated in Figure 17a, and the number of data words transferred on the bus, as shown in Figure 17c. There are several factors contributing to this disparity. The most significant is the use of a single remote terminal with only four subaddresses in the initial implementation

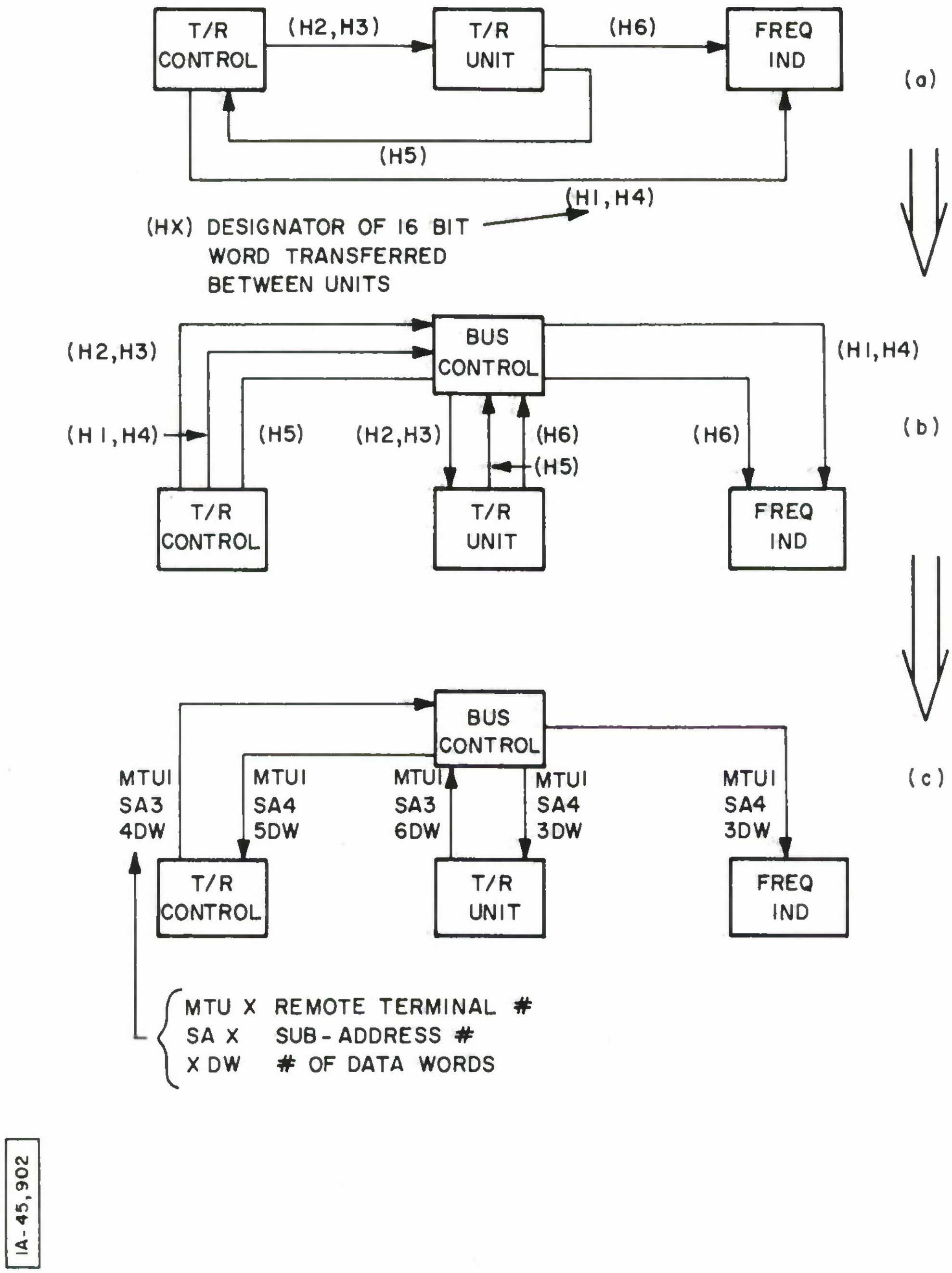


Figure 17 EVOLUTION OF INFORMATION FLOW TO MESSAGE FLOW FOR AN/ARC-50 UHF RADIO

of the demonstration. This necessitated the use of a common subaddress for data words intended for different units. For example, subaddress 4 is used for words to the T/R controller (H5), T/R box (H2, H3), and the frequency indicator (H1, H4, H6). Moreover, separate messages were used to each unit to permit easy changeover to the addition of a second MTU at a later stage in the implementation. While the flow of redundant data words incurred in the present case is somewhat artificial, resulting primarily from the unique conditions of the demonstration, analogous situations could arise in an operational system.

Applications Processing Sequence for AN/ARC-50 Data Words. The interface between the common response processing and the applications processing for the particular case of the AN/ARC-50 demonstration is shown in Figure 18a and 18b. All six words--the non-redundant subset of the AN/ARC-50 related data words received from the remote terminal--are required to be transferred to other units of the radio subsystem. In addition, four of the words must be processed by the bus controller to extract the information used in the AN/ARC-50 display and associated warning functions. The signals in these four words are formatted as shown in Figures 19a and b. As the common response processing sequences through each of the incoming data words, it initiates the processing sequences appropriate to their information content. The activities performed are largely self explanatory, and further discussion of the individual routines is not warranted. However, a general observation is worth making. The subsystem selected for the demonstration, the AN/ARC-50, was not orignally designed for use with a multiplex bus, and provides a good example of some of the problems involved in adapting "as is" equipment to bus usage. An example of this, in the area of applications processing, is the lack of uniformity in the representation of the information content in the data words; the digit 3 as coded in the frequency word is different from a 3 in the preset channel, which in turn differs from a 3 used to denote a power setting. Such lack of standardization necessitates the use of a number of software routines with identical functions but dissimilar in detail, resulting in a considerably larger software package than might be anticipated. When avionics equipment is designed specifically to operate with a multiplex bus, this will no longer be a problem.

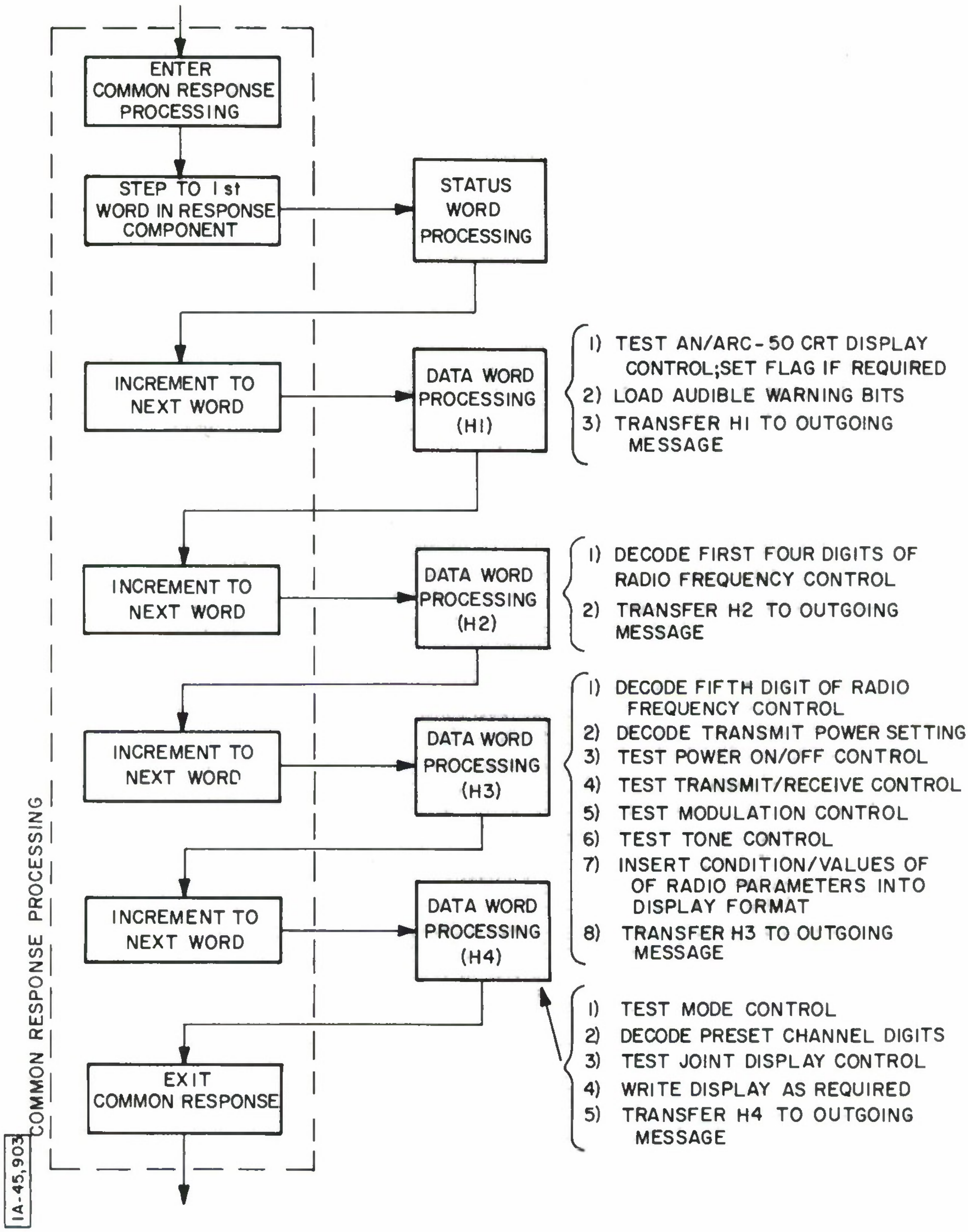


Figure 18a APPLICATIONS PROCESSING SEQUENCES FOR AN/ARC-50 DATA WORDS (MESSAGE M10102)

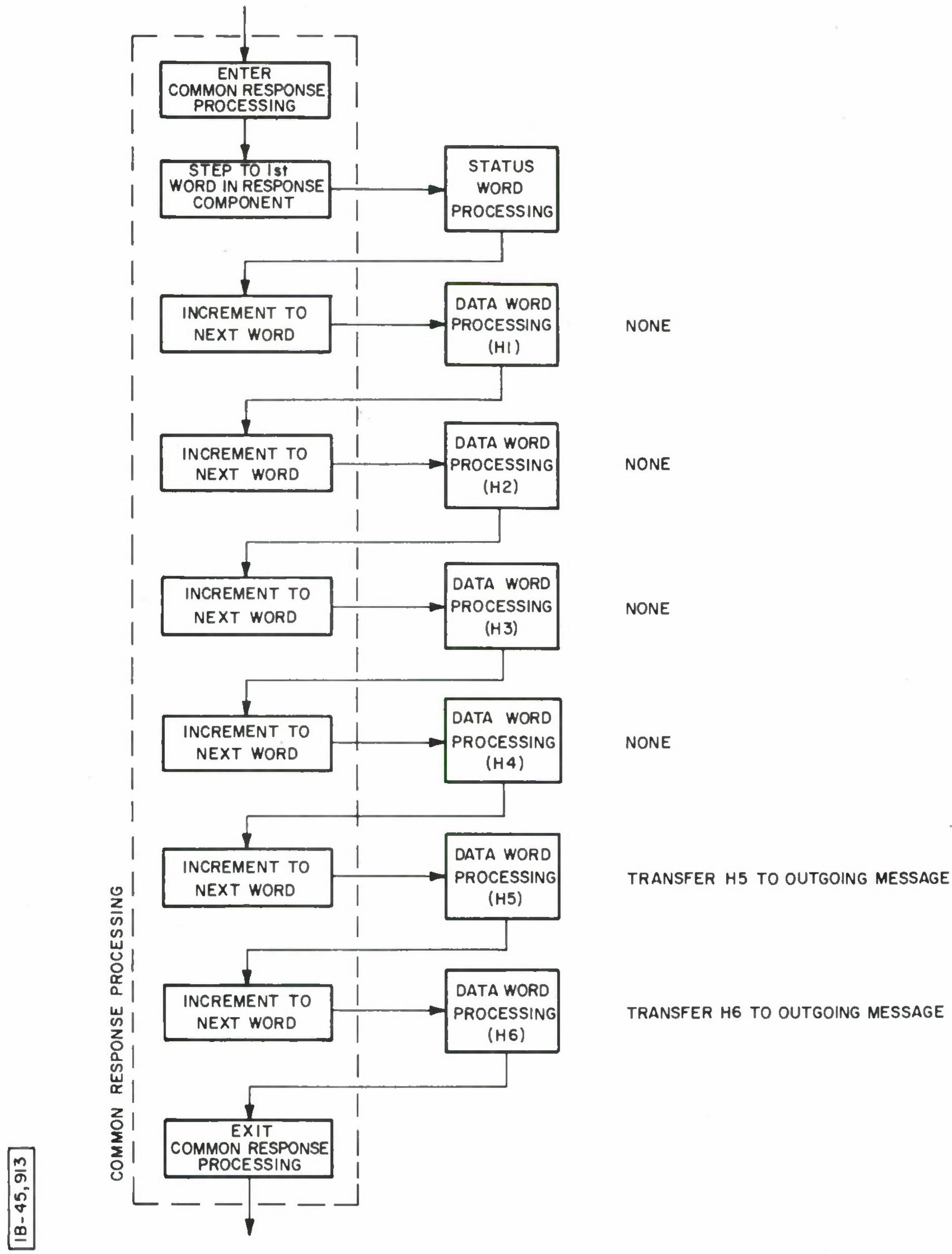


Figure 18:b APPLICATIONS PROCESSING SEQUENCES FOR AN/ARC-50 DATA WORDS (MESSAGE M10103)

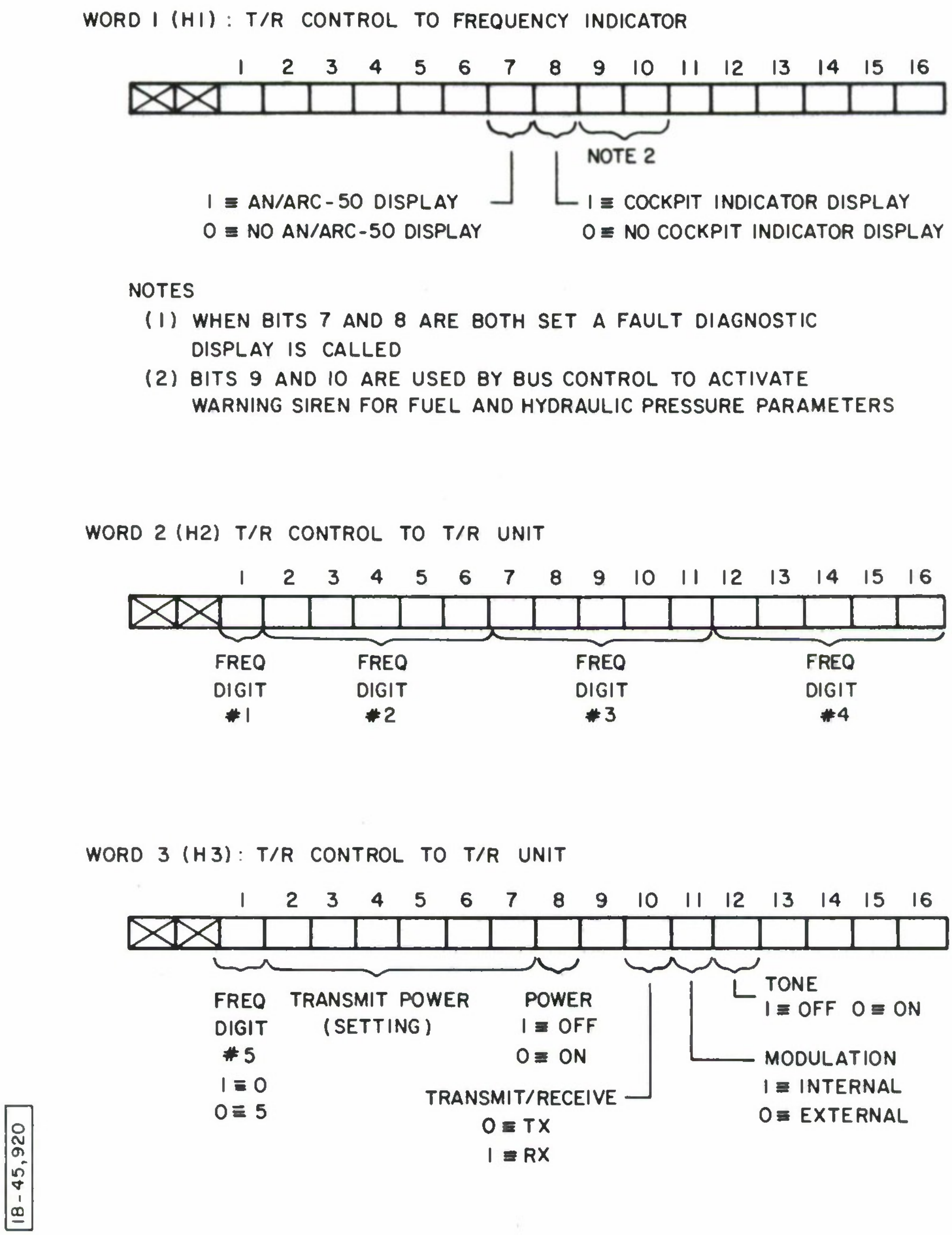
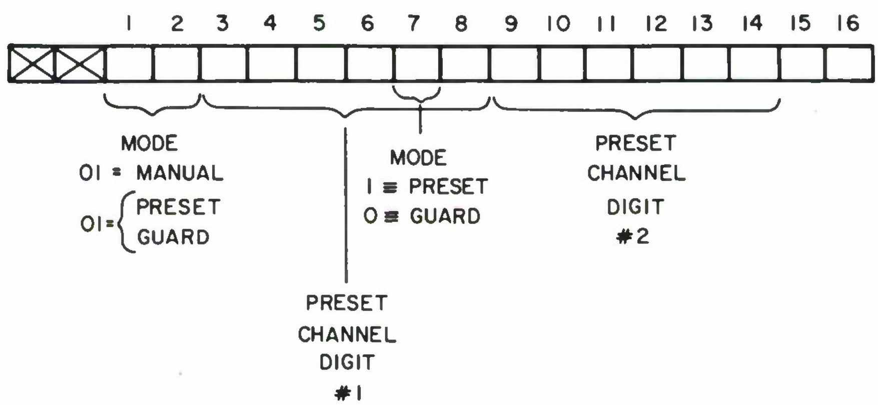


Figure 19a SIGNAL FORMATS IN AN/ARC-50 DATA WORDS

WORD 4 (H4) T/R CONTROL TO FREQUENCY INDICATOR



WORD 5 (H5) T/R UNIT TO T/R CONTROL

NO COMPUTER PERTINENT CONTENT

WORD 6 (H6) T/R UNIT TO FREQUENCY INDICATOR

NO COMPUTER PERTINENT CONTENT

1B-45,919

Figure 19b SIGNAL FORMATS IN AN/ARC-50 DATA WORDS

Outline of Applications Software for the Cockpit Indicator Demonstration

The approach to servicing the cockpit indicators by the multiplex bus is essentially identical to that used for the AN/ARC-50. However, the details differ somewhat.

Information Flow. The evolution of the MIL-STD-1553 version of the message flow between the cockpit indicators and their drivers from the information flow if dedicated wiring was used, is shown in Figures 20a to 20c. The same subaddresses, 3 and 4 of MTU1, that are used to service the AN/ARC-50 radio, are also used for the input and output data for the cockpit indicators. Since the message sequences for all the capabilities described in this report are loading the bus essentially simultaneously, and two words are required for the cockpit indicator data, eight words total will be used in subaddresses 3 and 4, i.e. six words for the radio, and two for the cockpit indicators. Two of the indicator drivers are sampled, and their data transferred to the indicators, four times per unit time, and one, the gyro repeater, twice per unit time; i.e. half and one quarter, respectively, of the sampling rate used for the radio control.

Applications Processing Sequence for the Cockpit Indicator Data Words. The interface between the common response processing and the applications processing for the cockpit indicators is shown in Figures 21a and 21b. The first six data words of the response component are not associated with cockpit indicators, and in consequence are not subject to applications processing at this time. The format and functional content of data words 7 and 8 are shown in Figure 22. The information transfer between the indicators and their drivers is in the form of binary numbers. The A/D converters associated with the fuel and hydraulic indicators quantize the driver (analogue potentiometer) settings into 8 bit words. The synchro converters produce 10-bit outputs. In all cases the processing consists of scaling the binary word according to the range of the parameter being represented, converting to 7 bit ASCII via BCD, and inserting the characters into a format for subsequent display on a Sanders 720 CRT.

Additional Demonstration Capabilities

In addition to the applications software associated with the UHF radio and indicator demonstrations, some simple fault diagnostic capabilities were programmed.

One of these is derived from the validation of the status word(s), which is a standard procedure on all response components. If the status word is valid, processing of the data words proceeds

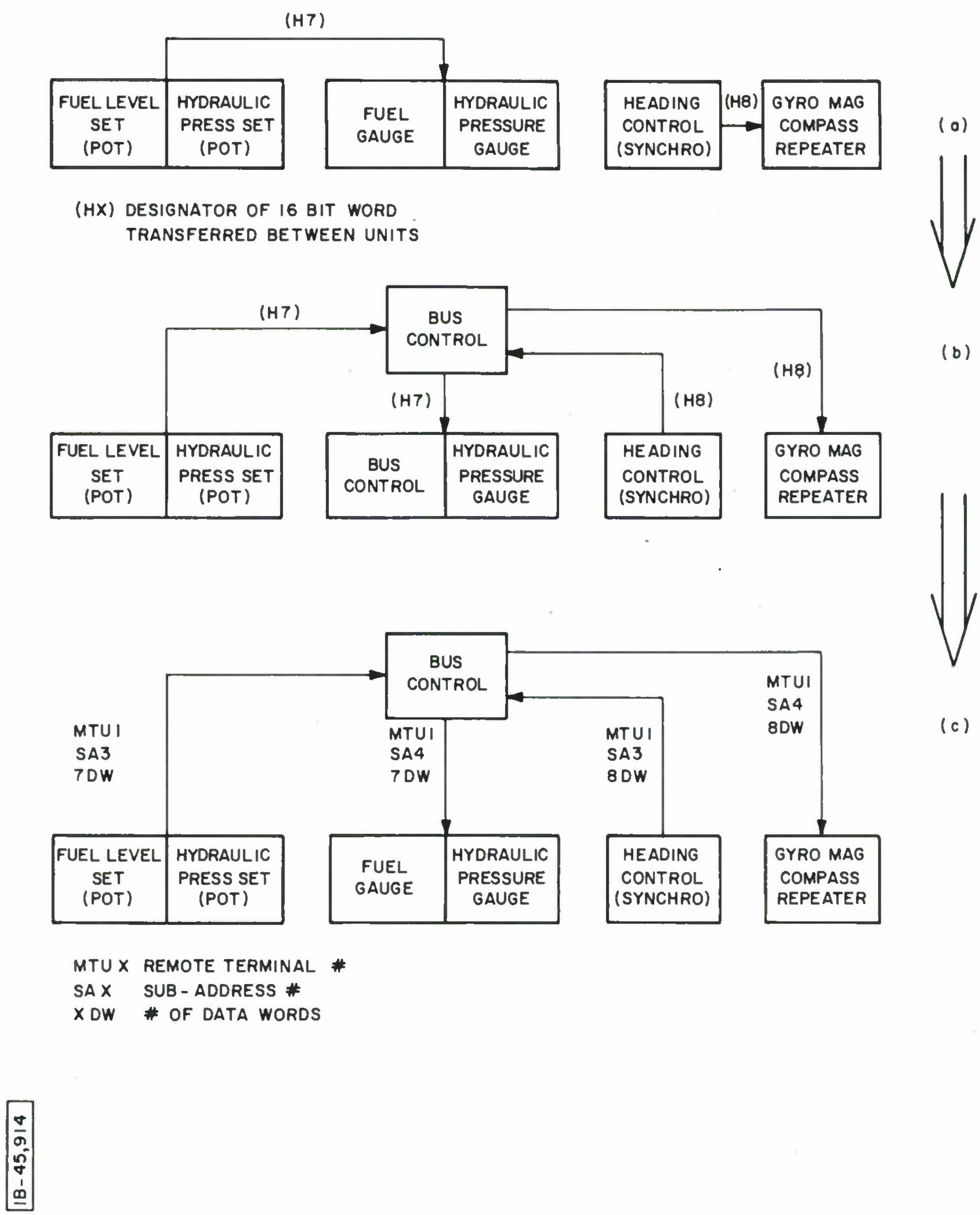


Figure 20 EVOLUTION OF INFORMATION FLOW TO MESSAGE FOR COCKPIT INDICATORS

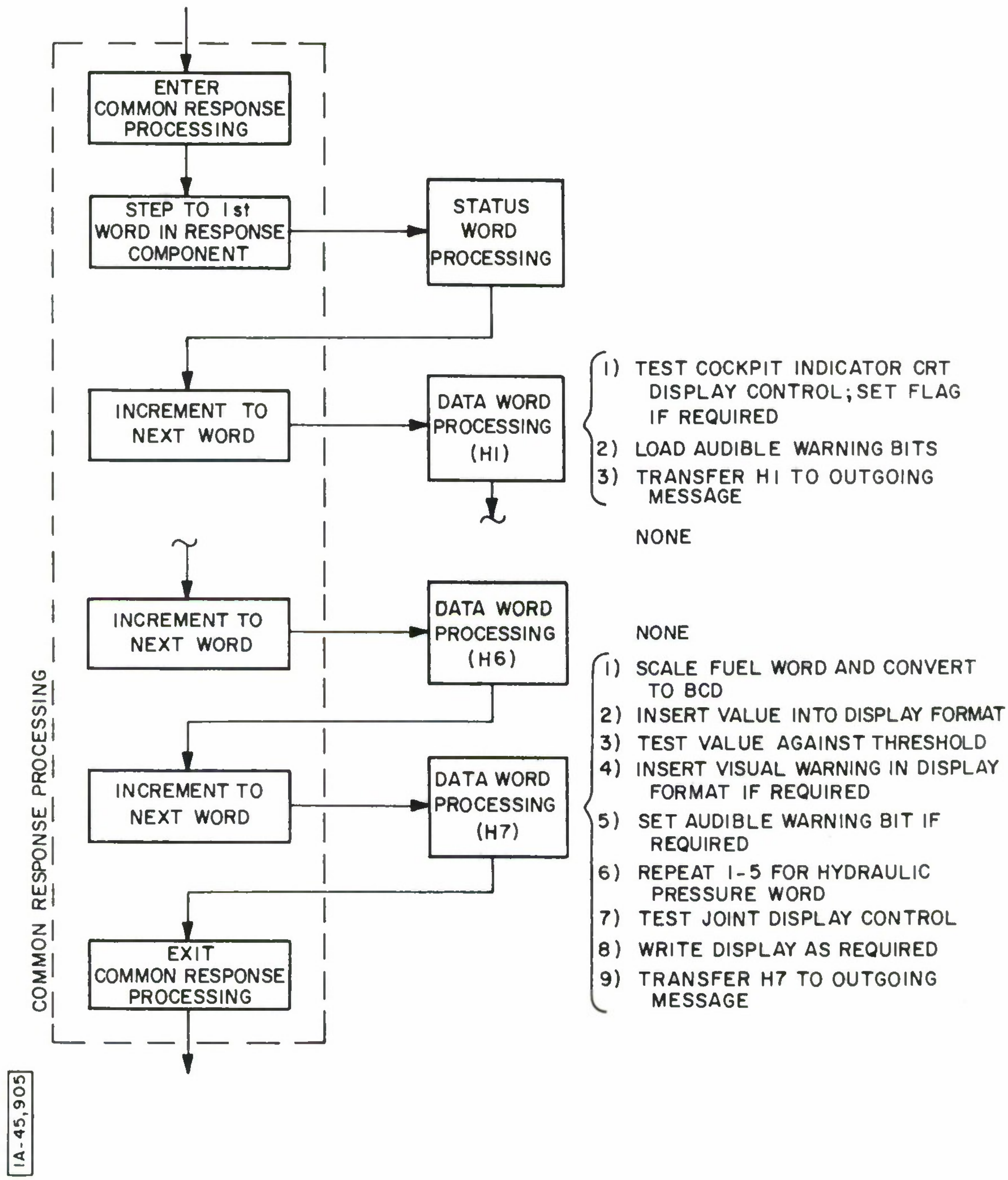


Figure 21:a APPLICATIONS PROCESSING SEQUENCES FOR COCKPIT INDICATOR DATA WORDS (MESSAGE M20102)

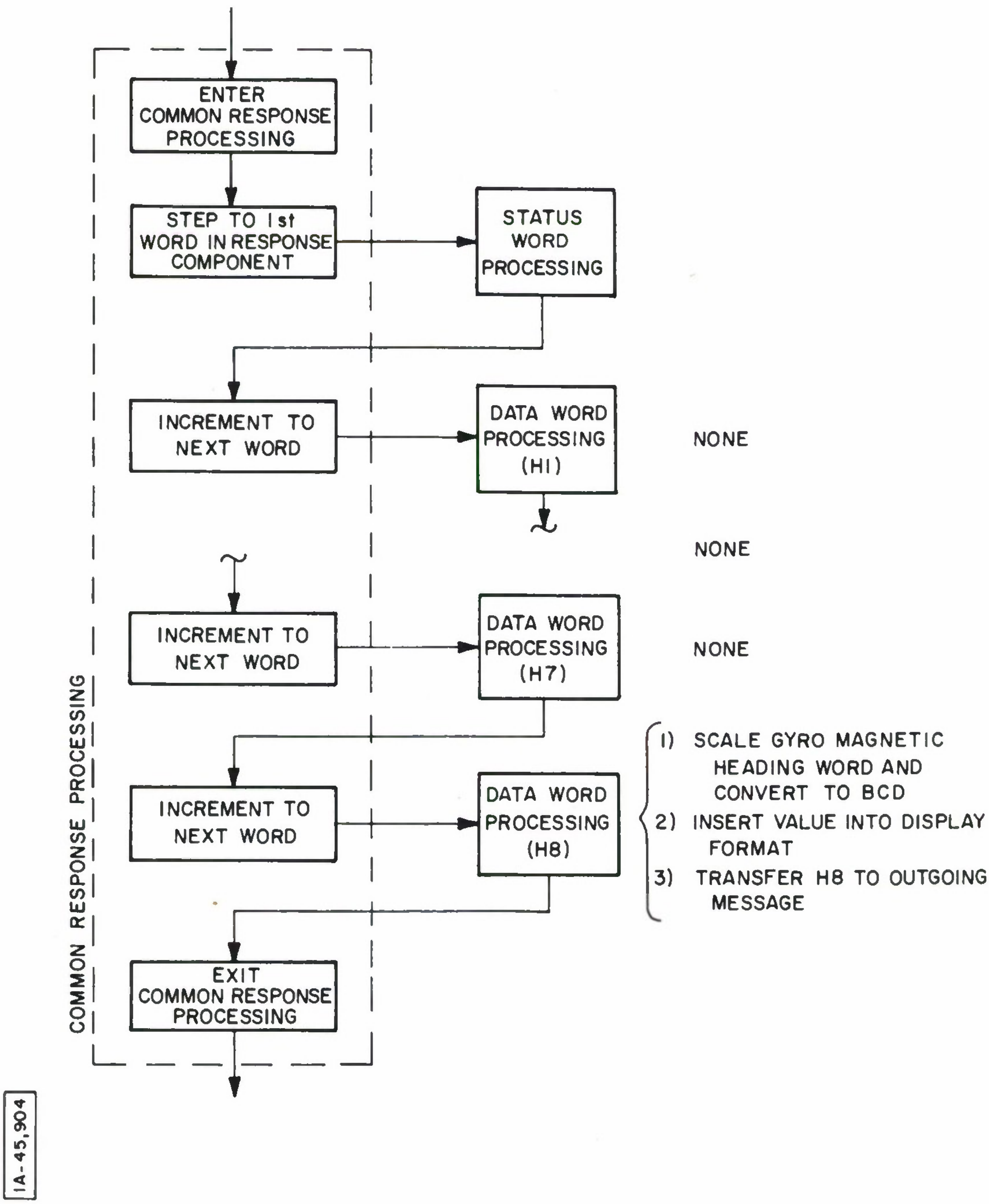


Figure 21:b APPLICATIONS PROCESSING SEQUENCE FOR COCKPIT INDICATOR DATA WORDS (MESSAGE M30301)

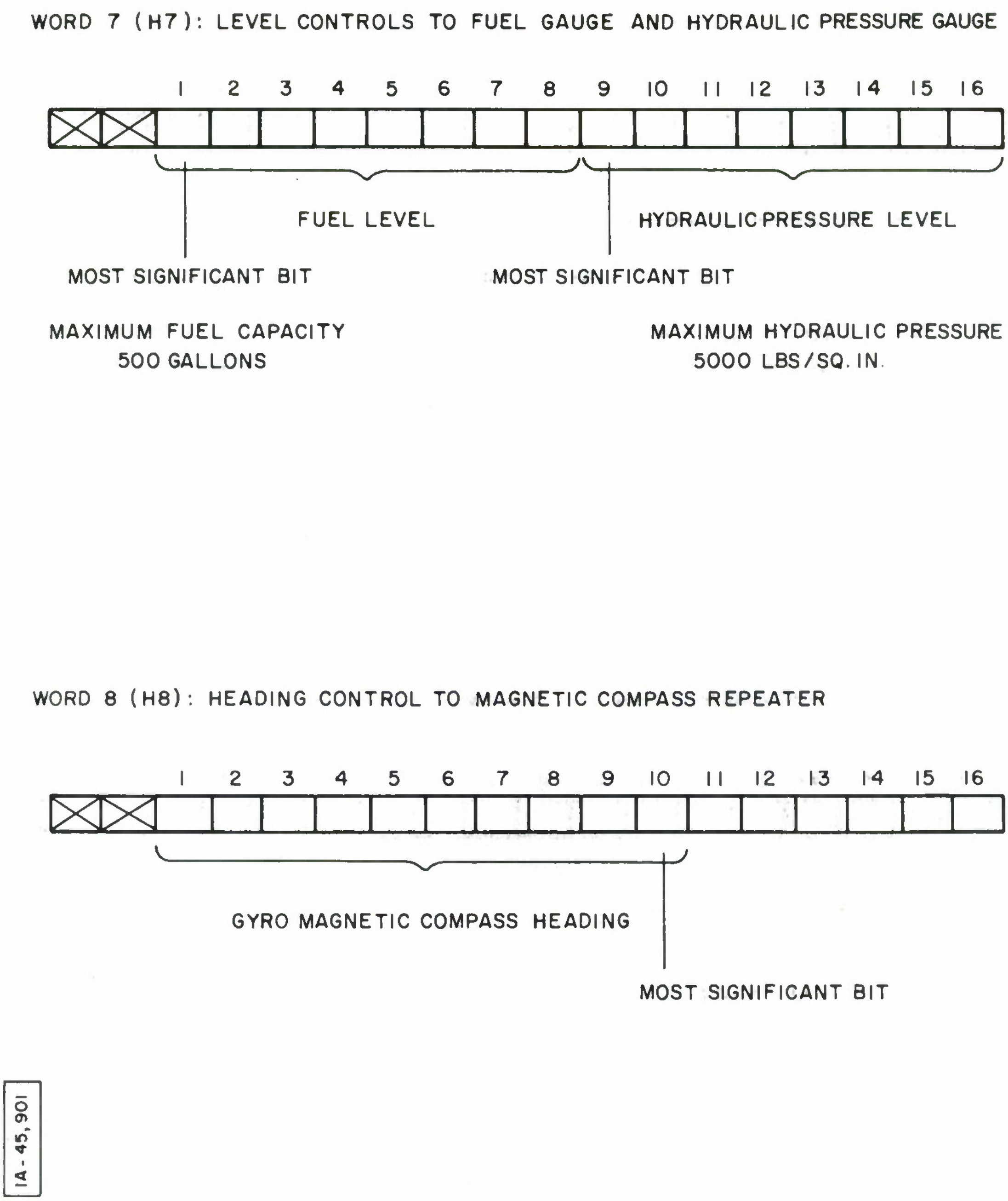


Figure 22 FORMAT AND FUNCTIONAL CONTEXT OF COCKPIT INDICATOR DATA WORDS

along the paths described previously. If, on the other hand, an error bit is set in the status word, the processing branches to a routine which prints out on the TTY the MTU address and subaddress to which the message had been sent. Subsequent applications processing of the data words in this response component is bypassed, and the message handler steps to the next message and proceeds routinely.

Another capability being demonstrated is the detection of operator errors and a display of the event. The operator is provided with two toggle switches which provide the following options: select AN/ARC-50 CRT display, cockpit indicator CRT display, or clear CRT display. A possible, but illegal, pair of switch settings calls both display formats in rapid alternating sequence. The applications processing contains a subroutine which interprets the bits, 7 and 8 in data word H1, which are set by the switches. If the illegal setting is detected, a "fault diagnostic" display is automatically called that informs the operator that the switch setting is incorrect.

The demonstration of the monitoring of flight parameters includes both the fuel and hydraulic pressure levels. The extraction of the latter from the incoming data words for display purposes has already been described in the previous section. The applications processing associated with the monitoring function is a simple extension to that operation. The level of the parameter, as determined from the data word, is compared with a preprogrammed threshold value. When the level falls below the threshold level, a routine is initiated which superimposes an intermittent DANGER signal adjacent to the critical parameter on the CRT display. In addition, a bit is set in an outgoing data word which activates an audible alarm at the remote terminal.

SECTION V

PROBLEMS ENCOUNTERED

General

During the course of designing, building, and testing the RIDS demonstration multiplex digital data system, several problems and constraints were encountered which impact on the system design and therefore warrant special mention in this document. These are described in this section.

MTU/SSIU Interface

MIL-STD-1553 (USAF) which has been issued as the Military Standard for Aircraft Internal Time Division Multiplex Data Bus Systems defines the interface between the MTU and SSIU. It specifies the type of data to be transferred (NRZ), the type of transfer (16 bits parallel bi-directional), the rate of transfer (1 MHz), and the control and status signals (command transfer, MTU data transfer, SSIU data transfer, SSIU status, and 4 MHz clock). Although the standard does not, per se, specify the details of the SSIU, the MTU/SSIU interface specifications do impose overly severe constraints on the design of the SSIU and do not permit sufficient latitude for optimizing circuit parameters. At the time this report is written, a revised MIL-STD-1553 is being considered for publication. This proposed revision does not specify the MTU/SSIU interface as the original standard did, so this will not be a problem if the revised standard issues in its present form.

Babbling

The standard specifies the capability for operating the multiplex data bus system in an austere environment with only a single (non-redundant) bus. It also specifies that all terminals will continually monitor the line and will not attempt to transmit when the bus is busy (i.e. another terminal is transmitting). These specifications give rise to the very real possibility (as has happened many times on the laboratory demonstration system) that one terminal will start to babble (i.e. transmit continuously) due to a malfunction. Then there is no way for the controller to command it to shut off. A redundant bus would at least permit access to the babbling terminal through a second port. Self checking circuits in the MTU to prevent this babbling condition from occurring should also be provided. However, for such a circuit to be useful, the self checking feature must itself be impervious to the malfunction which causes the babbling.

Stub Impedances

Adjusting the impedances of stubs in a multisubscriber time division multiplex bus design requires careful trade-offs. In order to minimize errors in data transmission, the designer would like a high signal to noise ratio. This requires low electrical noise in the usual sense as well as low pulse distortion caused by reflections and ringing of reactive circuits. MIL-STD-1553, therefore, specifies that the main bus must be terminated at each end in its characteristic impedance. With no stubs attached, the main bus would, under these conditions, look like a transmission line of infinite length with no disturbing reflections. A problem arises, however, when stubs are connected across the bus at various points to provide access to and from subscribers. The stubs load the bus locally, causing a mismatch with its accompanying reflections. To minimize this problem, the designer would like the stub to present an infinite impedance to the bus. Under these conditions, unfortunately, no signal energy would be transferred into the stub; so a first level trade off must be made between the receiver sensitivity requirements and the bus mismatch which can be tolerated. The lower the stub impedance, the more energy it will divert from the bus and the easier it will be to detect the signal. Other design goals, however, further complicate the problem. The subscriber not only receives information from the bus via the stub, he also transmits information onto the bus over the same stub. The designer would like to minimize his transmitter power requirements to reduce generator size, heating problems, and potential EMI radiation. The more subscriber receivers there are connected to the bus, the greater the transmitter power required to feed them. This is an argument in favor of sensitive receivers with minimal input power requirements. And finally, the standard specifies fault isolation resistors to be connected in each stub leg at the bus-stub junction (see Figure 2). This is to prevent a shorted stub from shorting out the bus. Unfortunately, the resistor pair in the transmitter stub represents a substantial signal power loss, and the pair in each receiver stub represent additional signal losses. Further trade offs must therefore be made between transmitter power requirements and receiver sensitivity.

This stub impedance problem was examined both experimentally in the lab and analytically with a computer simulation. The results indicate that the best waveforms and smallest transmitter to receiver losses are achieved not with the use of the transformer coupling as specified in the standard (although transformer coupling is possible), but with the use of stubs having a higher characteristic impedance than the bus, i.e. stubs made of 200 ohm characteristic impedance cable with the 70 ohm bus. This subject is covered thoroughly in four Technical Reports (see References 2, 4, 5, and 6).

Testing and Trouble-Shooting

Testing and trouble-shooting presented some unique problems in the checkout of the RIDS demonstration bus system. Several procedures have been developed which have materially shortened the development cycle and facilitated trouble-shooting. Initially, it was advantageous to connect the controller to a single remote terminal by means of a main bus (i.e. no stubs) for check-out of the circuits. The main bus was terminated properly by shunting the controller and remote terminal with resistors equal to the characteristic impedance of the bus. This procedure eliminated problems associated with reflections on the bus.

During initial circuit checkouts, the controller and remote terminal were dc coupled to the bus instead of transformer coupled as specified by the standard. This avoided the problems inherent in transformer coupling, and allowed the checkout to concentrate on circuit problems without having to cope with bus generated interferences.

One of the most useful test devices utilized in the check-out of the RIDS demonstration bus system is a flexible test software program which can be used in a number of modes. A "write only" mode transforms the controller into a specialized signal generator which repeatedly transmits commands to the RT under test without requiring a response. This permits step-by-step checkout of the RT circuit from the receiver input all the way to the transmitter output. A "read only" program causes the controller to transmit commands repeatedly to the RT asking for a given number of data words to be sent back. The controller can then read out the returned message for validation, but it does not hang up if the message is incorrect; it continues to send out the command at fixed intervals which are sufficiently long to permit the remote terminal to respond.

SSIU to Subsystem Interface

The recent rapid advances in LSI technology seem to indicate that a complete remote terminal (with the possible exception of the power stages of the MTU transmitter) can be fabricated on a single (or at the most a very few) circuit chip(s). According to MIL-STD-1553, each RT must be capable of distributing data to as many as 32 subsystem addresses and must be able to collect data from as many as 32 subsystem addresses. If each of these channels were to be provided with a dedicated interconnecting cable between the RT and the avionics as suggested in the standard, the remote terminal would have to be equipped with 64 multipin connectors just to interface with the avionics subsystems. The degree to which connectors can be miniaturized is limited by the size of the human hand which must manipulate the connections. It is readily apparent that 64 connectors of even the smallest practical size will require far more space than all of the circuitry in the RT. This clearly becomes a case of the tail wagging the dog. Even in the RIDS laboratory demonstration, where no particular effort was made to miniaturize any of the equipment, the space required for connectors presented difficulties. This problem could be alleviated by time division multiplexing all data between the RT and the associated avionics subsystems over a single bus such as the DEC Unibus or the proposed International Electrotechnical Commission Instrument Bus. This would require that the signal conditioning and level conversion functions be removed from the SSIU and relocated at the individual avionics equipment. In the future, when equipment is designed to connect to a multiplex bus, this will not be a problem. However, it does represent a problem when avionics in the current inventory are used. Nevertheless, this appears to be the only way to take advantage of the potentially small size of the RT circuitry.

Software Constraints

There are some constraints which have been imposed by MIL-STD-1553 that gave rise to difficulties in the development of the message control software for the RIDS experimental bus. These are described briefly in this subsection; for a thorough discussion see Reference 7.

Upper Bound on Bus Capacity

The message protocol coupled with the word formats specified in the standard result in a relatively high overhead in terms of bus capacity. Figure 23 presents a graph showing the percentage of bus capacity available for information transfer versus the number of data words in a command/response message sequence. It can be seen from this graph that the efficient utilization of the bus capacity is markedly dependent on the number of data words in a message. It should be noted that the values plotted assume that the 16 information bits in each data word carries useful information. While in concept this does not appear difficult to achieve, in practice its realization can lead to considerable sophistication in both the message control software and in the bus hardware.

Time Constraint on Message Control Processing

When considering the message handling functions to be performed by the bus controller, it becomes apparent that the message rate on the line is a more significant parameter than the data rate. This arises primarily because the majority of operations in the message handling cycle are independent of message length; thus a given data rate with short messages results in a completely different processor loading than the same data rate with long messages.

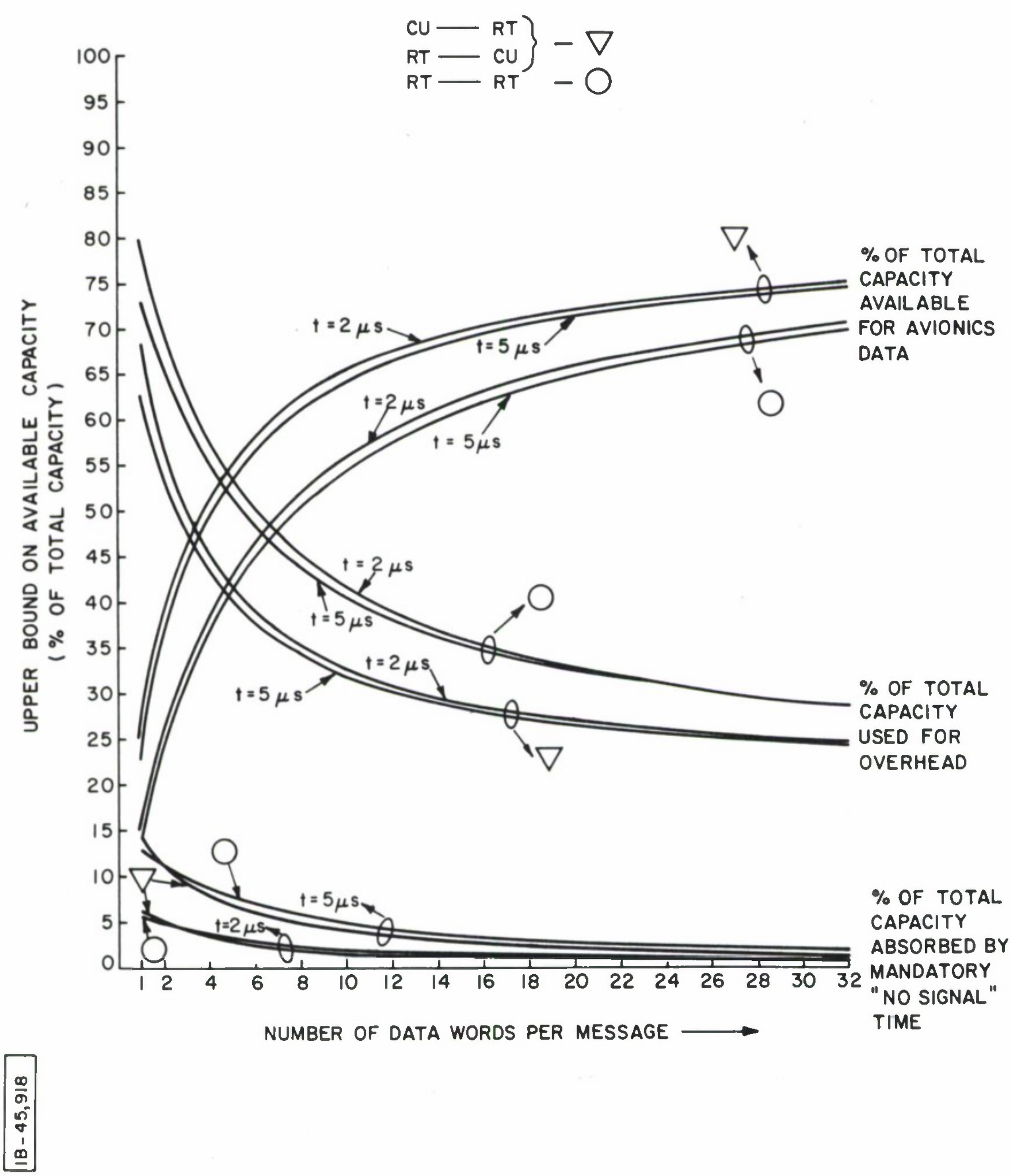


Figure 23 "STANDARD" MUX BUS: AVAILABLE INFORMATION CAPACITY

The message protocol defined in the standard permits a range of message lengths. At the lower end of spectrum, the combined command/response message sequence consists of only three words: command word, status word, and one data word. Since the data rate on the line is required by the standard to be 1 megabit per second, the duration of such a message is approximately 65 microseconds. It can be seen, therefore, that operation of the bus at a high duty cycle with very short messages would require that the bus controller complete the basic message cycle in excess of 10,000 times per second. This would present a formidable load for a general purpose airborne computer, and would even necessitate considerable sophistication in a special purpose machine.

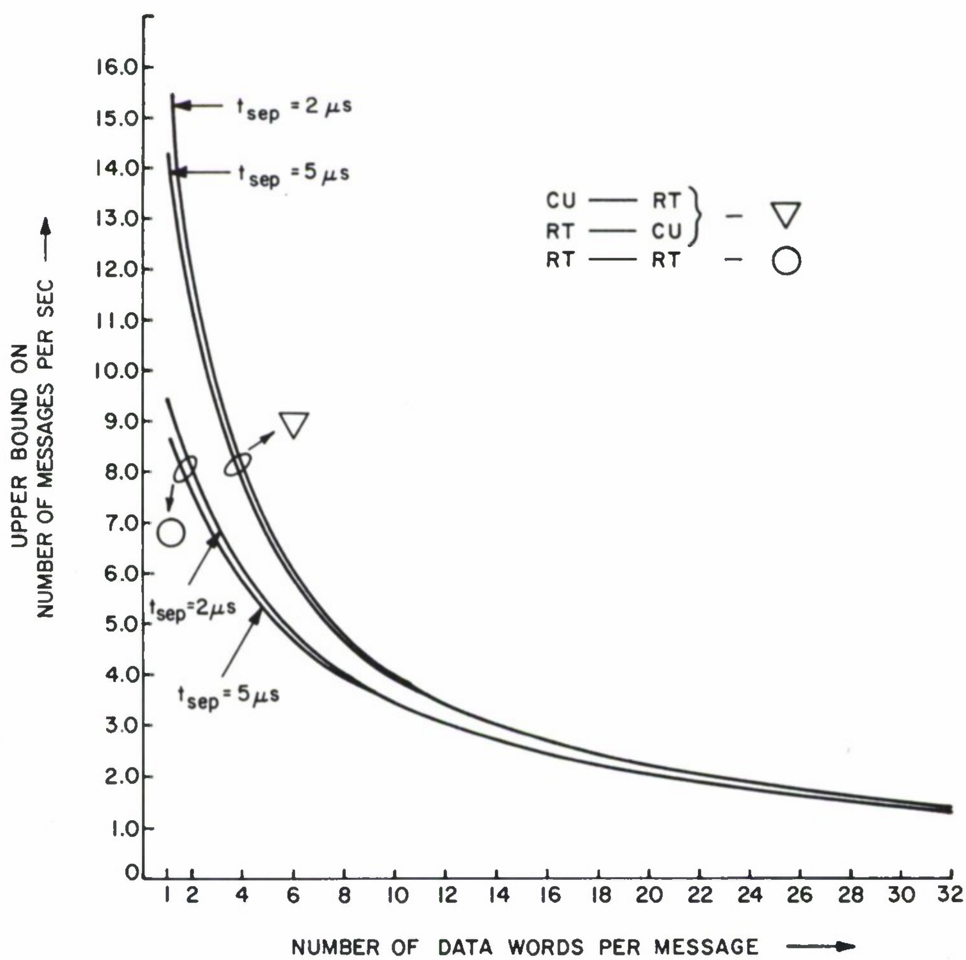
It may be argued that such a situation would never arise. This might well be true, but it is the function of a standard, when defining a system tool, to provide a user with an appreciation of what capabilities are required. If the above capabilities are not considered reasonable, even though they are implied in MIL-STD-1553 (USAF), then some additional limitation on the message handling requirements should be specified.

Message and Word Packing

The considerations of bus utilization and the limited execution time for the message handling cycle that have been mentioned above, point to the desirability of increasing the information density within a message. The standard permits two ways of accomplishing this increase: message packing and word packing.

It can be seen from the graph in Figure 24 that if the number of data words in a message can be increased, the fractional capacity of the bus available for signal transfer between subsystems serviced by the bus can also be increased. Since, in many cases the information generated by a single avionics subsystem can be encoded into a very few data words, the increase in message length can be achieved only by combining into the same message the outputs from several subsystems that are located at the same RT. This process is known as message packing.

The technique of message packing results in more data words per message, and thus in a higher upper bound for available information capacity of the bus. However, it does not guarantee that the information density of the message will be increased; this will happen only if the information content of each data word is dense. Experience has shown that in many cases the signals generated by a source subsystem do not require all 16 bits in a data word for their representation. In extreme cases such as discrete signals, a single bit is sufficient for transmission of the information. The dedication of a whole 16-bit data word for each signal could thus



IB-45,916

Figure 24 NUMBER OF MESSAGE SEQUENCES PER SECOND result in operating the bus way below its capacity. To offset this inefficiency, different signals generated by the same subsystem, or by several subsystems at the same remote terminal, can be combined into a single data word. This process is known as word packing.

These two techniques, which are conceptually so innocuous, have the potential for significantly impacting several aspects of the bus design. These range from drastically increased execution time for the basic message handling cycle to incompatibility of buses built to the same standard. A more detailed discussion of this subject is presented in Reference 7.

SECTION VI

RESULTS

RIDS Demonstration Bus Design

The most important result of the work performed under Project 6370 is that a demonstration time division multiplex digital data bus system has been designed and built in accordance with the requirements specified in MIL-STD-1553 (USAF). This bus system has been operating since October 1974. As far as is known, this is the only MIL-STD-1553 type bus system presently in operation.

The work performed in the design, fabrication, and check-out of both the hardware and software for the RIDS demonstration bus system has provided a wealth of experience in the problems associated with the design and operation of such a system, and has provided the opportunity to investigate various aspects of the 1553 standard in order to determine feasibility. These investigations have brought to light several problems and undesirable constraints associated with MIL-STD-1553 (USAF) that were described in Section V.

Software Programs for Component Tests

MITRE has developed a series of test procedures and software programs to facilitate the check-out of various bus system components. These software programs, in effect, permit the use of the bus controller as a special purpose signal generator which can transmit onto the bus various command words in the proper format to elicit different types of responses from the remote terminal under test. The controller repeatedly transmits the command words at fixed intervals without hanging up if the response is incorrect. This permits monitoring of the remote terminal functions, since the command signal repetition rate is high enough to permit observation with an oscilloscope. Different command words exercise different remote terminal functions. After individual RT functions are tested and corrected if necessary, different software programs permit testing several functions together, and then finally the system as a whole. This program also serves as a fast means of insuring that the system is "up and running."

Demonstration

The operational RIDS multiplex data bus system has provided a tool for demonstrating to various interested groups the basic concepts of such a system. The inherent flexibility and ease of system reconfiguration have been demonstrated; the fact that existing USAF inventory avionics hardware can be made to work in the system has also been proven.

Investigation of the Transmission Medium

A new method of connecting stubs to a multisubscriber bus was investigated through both computer simulation and laboratory experimentation. (See References 2, 4, 5, and 6.) This method was shown to produce better waveforms, significantly lower signal power losses, and does not require any stub-to-bus coupling transformers. A recommendation was made to modify MIL-STD-1553 to permit this type of stub coupling.

Inputs to Standards Committees

As a result of work done and experience gained in the design, fabrication, and testing of the RIDS demonstration bus system, it was possible to provide comments to the Standards Committee at AFAL that is responsible for producing MIL-STD-1553 (USAF). Often analytical studies were made to provide a basis for recommendations (e.g. Reference 8). The proposed "General Specification for an Aircraft Multiplexed Data Bus," generated by the Tri-Service Multiplex Bus Committee, was also reviewed. This proposed Tri-Service Standard was based on MIL-STD-1553, but greatly extended the operational concept. MITRE submitted detailed comments to this committee also.

Use of Microprocessor in SSIU

The RIDS demonstration bus system has been provided with two remote terminals (see Section II above). The SSIU in one of the terminals has a hardwired multiplexer for the distribution of data to and from the avionics subsystems. The second remote terminal SSIU, however, has been provided with a microprocessor which performs the multiplexing function. This microprocessor is a programmable general purpose unit which greatly increases the SSIU flexibility and capability in permitting reconfiguration through software changes and in performing such data processing functions as word and message packing and unpacking, data correlation, and data routing, etc. Although MIL-STD-1553 (USAF) does not call for a microprocessor in the SSIU, the work in this area has demonstrated its worth, and most people concerned with multiplexed digital data bus systems are now convinced that using a microprocessor in the RT is advantageous.

Papers Presented at Engineering Conferences

As a result of work done on the RIDS program, seven technical papers have been presented at various engineering conferences in the

United States and Europe. These papers have described state-of-the-art technology and have been favorably received. They are listed in References 9 through 15. The NAECON papers were reprinted in Reference 16.

Technical Reports

As shown by the substantial number of MITRE Technical Reports in the references, the program has been well documented. In addition to these, nine working papers were prepared to disseminate information only within MITRE. Another related document prepared under Project 6540 is shown in Reference 17.

SECTION VII

RECOMMENDATIONS

As a result of the analysis, simulation, and laboratory experimentation which has been done under the RIDS program, the results of some of the design choices are now clear. In this section a set of recommended choices are stated with a brief indication of the reasons for the choices.

High Characteristic Impedance Cable

Both simulation and laboratory experimentation have shown that coupling the stubs of a multisubscriber bus into the main bus can be achieved in several ways. Although MIL-STD-1553 (USAF) does not at the moment permit the use of a high characteristic impedance cable for the stubs, this has proved to be the best means of coupling the stubs which we know. The high characteristic impedance cable provides better waveforms, lower transmitter to receiver losses, and does not require the stub-to-bus coupling transformer. Other organizations should attempt to duplicate the results which have been obtained under Project 6370 and should consider using this design in future systems.

Remote Terminal and BCIU Design

The proposed revision of MIL-STD-1553 no longer stipulates the interface between the MTU and the SSIU. This will permit the designer much greater freedom. Based upon our experience using a microprocessor as a part of the RT, we recommend that the designer consider microprocessors, direct memory access, and newer components such as First-In-First-Out devices (FIFOs) and Content Addressable Memories (CAMs).

A Bit Parallel, Word Serial Bus

Many of the remote terminal functions can be achieved by using a microcomputer. In particular, many of the bus interface functions and the control and timing functions can be performed through software. The signal conditioning portion of the remote terminal will ultimately be accomplished in the avionic subsystems when subsystems are designed to connect directly to a digital multiplex bus. Therefore, the remote terminal will eventually consist of low level logic circuits of the approximate complexity of a microcomputer. The exception may be the power output stages of the transmitter. One would expect that the entire remote terminal will then be available on one or at most a few chips. These should fit into a box about 3 by 4 by 5 inches. However, each remote terminal can service up to 64 subsystems if each subsystem is allotted one subaddress. If a separate cable must transfer information to and from each of these subsystems, the number of leads will run into the hundreds, and it will be impossible to put the connectors on a box of the size noted above. One way to avoid this difficulty is to use a bit parallel, word serial bus of the Unibus type used by Digital Equipment Corporation or the proposed International Electrotechnical Commission Instrument Bus to convey information to and from the subsystems. We believe that it is imperative that the size of the RT be made as small as possible, and the secondary bus is a necessary consideration toward achieving this goal.

Software

The software programs which have been developed for testing and debugging have proven so useful that it is recommended that these programs be made a part of the total software package. The availability of this type of test software will greatly minimize checkout time and maintenance training time.

Changes in MIL-STD-1553

Project 6370 conclusions and recommendations have been made available to the multiplex bus committee at Wright-Patterson AFB through the project. Many of the recommendations have been incorporated in the original standard and the proposed revision. However, there are still a few changes which we believe should become a part of this standard. These are stated in the following paragraphs.

High Characteristic Impedance Cable

The use of a high characteristic impedance cable for impedance matching the remote terminal to the main bus should be permitted. The need for a coupling transformer should be removed.

Stub Impedance Presented to the Main Bus

The current standard requires that the stub impedance presented to the main bus be greater than 2,000 ohms at 1 MHz. The power loss from transmitter to receiver is minimized when the stub presents an impedance of about 750 to 1500 ohms. The standard should be changed to reflect this.

Duty Cycle

The standard should be ammended to specify a minimum duty cycle for information transfer on the main bus which the equipment can maintain. No duty cycle is currently specified, yet the designer of the RT and the BCIU will make greatly different design choices when the minimum duty cycle is 20 percent or 80 percent. The standard should be modified to present guidance to the designer in this respect.

Transmitter Power

The standard should be modified to permit the transmitter power to be measured under known and standard conditions. Currently the transmitter power is measured into an unspecified bus at a point remote from the transmitter and it is difficult for the transmitter designer to establish a reasonable manufacturing test.

Addresses

The standard permits the addressing function to use 11 bits: five for the terminal address, five for the subaddress, and one to determine whether the remote terminal will transmit or receive. If the functions of the remote terminal can be obtained in a single LSI chip, it should be economically feasible to place a remote terminal in nearly every subsystem. Then it might be desirable to have many more terminals and fewer subaddresses at each terminal. The standard could and should permit this addressing flexibility at the discretion of the system designer without increasing the total number of bits devoted to addressing.

REFERENCES

1. "Aircraft Internal Time Division Multiplex Data Bus," Military Standard 1553 (USAF), 30 August 1973.

2. The MITRE Corporation, "Radio Information System Architecture," ESD-TR-75-70, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, Mass., July 1975.

3. P. R. Cloud, "An Outline of the Applications Software for the RIDS Experimental TDM Bus," ESD-TR-75-78, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, Mass., September 1975.

4. B. B. Mahler, "An Analytical Examination of Impedance and Signal Power Levels Along a Multiplex Data Bus," ESD-TR-75-52, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, Mass., March 1975.

5. R. A. Costa, "Computer Simulation of MUX Bus Voltage Waveforms Under Steady State Conditions," ESD-TR-75-67, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, Mass., June 1975.

6. E. F. Boose, R. A. Costa, and B. B. Mahler, "A Comparison of Time Division Multiplex Data Buses Based on Computer Simulation," ESD-TR-75-60, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, Mass., May 1975.

7. P. R. Cloud, "The 'Standard' MUX Bus: Some Constraints on Information Flow Arising from MIL-STD-1553 (USAF)," ESD-TR-75-96, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, Mass., November 1975.

8. Dr. J. Maeks, "Multiplex Bus Data Waveform Spectra," ESD-TR-75-64, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedofrd, Mass.

9. C. R. Husbands, "Use of Intelligent Terminals in Digital Avionics Multiplexed Data Transmission Systems," ACM Computer Science Conference, Detroit, Michigan, February 1974.

10. E. F. Boose and C. R. Husbands, "Application of Microprocessors in Avionics Systems," Journess d"Electronique 1974, Lausanne, Switzerland, October 1974.

11. C. R. Husbands, "The Application of Microprocessors to Low Data Rate Information Distribution Systems," ACM Computer Science Conference, Washington, D.C., February 1975.

12. R. A. Costa, "Computer Simulation for Multiple Subscriber TDM Networks," ACM Computer Science Conference, Washington, D.C., February 1975.

13. E. F. Boose, "Lessons Learned Through a MIL-STD-1553 Time Division Multiplex Bus," NAECON '75, Dayton, Ohio, June 1975.

14. C. R. Husbands, "Microprocessors in Airborne Time Division Multiplex Terminals," NAECON '75, Dayton, Ohio, June 1975.

15. E. F. Boose, R. A. Costa, and B. B. Mahler," Shielded Twisted Pair as a Transmission Medium in a Multisubscriber Time Division Multiplex System," NAECON '75, Dayton, Ohio, June 1975.

16. E. F. Boose, R. A. Costa, C. R. Husbands, and B. B. Mahler, "RIDS Papers for NAECON '75," MTR-3023, The MITRE Corp., Bedford, Mass., May 1975.

17. E. F. Boose, "Radio Subsystem Interfaces," ESD-TR-74-196, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, Mass., November 1974.